Handheld synthetic array – Final report, Part A

Defence Research and Development Canada

FINAL REPORT PART A

MARCH 31, 2014

for

Department of National Defence

DRDC-Ottawa

contract number w7714-115195/A

Department of Electrical and Computer Engineering

The University of Calgary

2500 University Dr, N.W.

Calgary, T2N 1N4

Executive Summary

This document is the compilation of the analysis and results of the research contract w7714-115195/A entitled Research & development for Long Term Evolution (LTE) wireless location based on synthetic array. The report is a compilation of the research activities and outcomes for the period of January 2012 to March 31 2014.

The report consists of two parts. This document is part A and part B is a separate document.

Acronyms

AN	Access node (eNodeB)
AP	Anchor point
ANan	Access node that is an anchor node
ANfp	Access node that is feature point
AWGN	Additive White Gaussian Noise
AOA	Angle of Arrival
BCRLB	Bayesian Cramer-Rao Lower Bound
BF	Bayesian Filter
BFI	Bayesian Fisher Information
BFIM	Bayesian Fisher Information Matrix
CDMA	Code Division Multiple Access
CML	Concurrent Mapping and Localization
CRLB	Cramer-Rao Lower Bound
CV	Computer Vision
EKF	Extended Kalman Filter
FastSLAM	Factorized Solution for Simultaneous Localization and Mapping
FI	Fisher Information
FIM	Fisher Information Matrix
FP	Feature Point
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
i.i.d	Independent and Identically Distributed
IMM	Interacting Multiple Model
IMU	Inertial Measurement Unit

IEEE	Institute of Electrical and Electronics Engineers
KF	Kalman Filter
KL	Kullback-Leibler
LAMBDA	Least-squares Ambiguity Decorrelation Adjustment
LJG	Linear and Jointly Gaussian
LML	Local Maximum Likelihood
LOS	Line of Sight
LTE	Long Term Evolution
MAP	Maximum A Posterior
MC	Monte Carlo
M-CRLB	Modified Cramer-Rao Lower Bound
MEMS	Micro Electro-Mechanical Systems
MI	Mutual Information
MLE	Maximum Likelihood Estimator
MM	Multiple Model
MMSE	Minimum Mean Square Error
MN	Mobile Node
NEES	Normalized Estimation Error Squared
OCXO	Oven-Controlled Crystal Oscillator
OWLS	Opportunistic Wireless Localization System
PDF	Probability Density Function
PDOA	Phase Difference of Arrival
PEB	Position Error Bound
PF	Particle Filter
PRN	Pseudo-Random Noise

POA	Phase of Arrival
RB	Rao-Blackwellized
RBPF	Rao-Blackwellized Particle Filter
RFID	Radio Frequency Identification
RIM	Research in Motion
RTOF	Round-trip Time of Flight
SLAM	Simultaneous Localization and Mapping
SNR	Signal-to-Noise Ratio
SS	Signal Strength
TCXO	Temperature Compensated Crystal Oscillator
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UWB	Ultra Wideband
WiFi	Wireless Fidelity
WLAN	Wireless Local Access Network

Variables used

variable	description
$bel\left(x_{0:t}\right)$	Posterior pdf of the particles over the complete time interval from 0 to tT_s
b_t	range offset
B_s	signal bandwidth
c	propagation velocity
$e_{1\times m}^{i}$	$1\times m$ row vector whose elements are zero, except the i-th element which is equal to one.

f	carrier frequency
f_b	clock frequency bias
$\bar{f}$	clock frequency drift
$f_r(t)$	clock random frequency error
$I(X,Y)$	mutual information between Y and X
J	Fisher information matrix in general
J_{tot}	total Bayesian Fisher information matrix
J_Z	measurement information matrix
J_P	apriori information matrix
$\mathbf{K}_{AN}$	vector of direction
$\mathbf{m}$	stacked vector of AN locations
$\mathbf{m}_i$	state vector describing the location of the ith ANs
$\mathbf{m}_{ap}$	stacked vector of APs location
$\mathbf{m}_{fp}$	stacked vector of FPs location
$\left\{ m_{i,x}, m_{i,y} \right\}$	2D location variables of an AN
N_{AN}	number of ANs
N_{AP}	number of APs
N_{FP}	number of FPs
N_d	dimension of dynamic variable in state vector
N_s	dimension of stationary variable in state vector
N_p	number of particles
$\mathbf{p}_t$	MN location vector
$\mathbf{p}_{1:t}$	history of MN locations
$\mathbf{q}_t$	state vector at time step t

$\mathbf{q}_{d,t}$	dynamic portion of state vector at time step t
$\mathbf{q}_s$	stationary portion of state vector
Q_d	covariance matrix of dynamic variables update process
Q_P	motion process covariance matrix
$r(t)$	geometric range between the AN and MN
$s(t)$	bandpass signal
$\tilde{s}(t)$	lowpass signal
$\mathbf{u}_t$	control vector
$\mathbf{u}_{1:t}$	history of update control inputs
$\mathbf{v}_t$	motion update process noise
$w_{i,t}$	measurement noise
$\mathbf{z}_t$	measurement vector received at time step t
$\mathbf{z}_{i,t}$	observation vector from i-th AN at the time t
$\mathbf{z}_{i,1:t}$	history of observation from i-th AN
$\delta_{k+1}^{i,j}$	$(k+1)\times(k+1)$ dimensional matrix whose elements are all zero except at the i-th row and the j-th column which is one
δt_{MN}	MN clock bias
δt_{AN}	AN clock bias
$\dot{\delta t}_{MN}$	MN clock rate drift
$\dot{\delta t}_{AN}$	AN clock rate drift
$\Delta\varphi(t)$	carrier phase variation
Δ_α^β	$\nabla_\alpha \nabla_\beta^T$, second order derivative operator
ε_φ	carrier phase measurement noise
η	normalizing constant

λ	carrier wavelength
$N(b, B)$	Gaussian process with the mean vector b and covariance matrix B
$\prod$	overall transition matrix of sight states
$\prod_i$	transition matrix of the *i*-th AN's sight state
$\sigma_b^2(Q_b)$	range offset variance (covariance matrix)
φ_0	carrier phase of AN transmitter at time of transmission
$\varphi(t)$	partial carrier phase cycle measurement
χ^2	chi-square distribution
$\dim(.)$	dimension function
$\mathrm{Re}\{.\}$	real part
$\nabla.$	Jacobian function

Outline of Part A

Chapter 1 Overview of contract objective, phases and deliverables

1.1 Overview of the developed solution of SGL

1.2 Outline of the report

Chapter 2 Overview of SLAM SGL Algorithm

2.1 Overview

2.2 SLAM information filter

2.3 LTE downlink signalling

Chapter 3   CV observables and processing

3.1 Overview

3.2 System Definitions

3.3   2D translation ego-motion

3.4. Rotation of the World axis

3.5. General affine transformations

3.6 3D Perspective ego-motion based on a 2D marker and unknown fp`s

3.7 Alternative 3D Method

3.8 Experimental results of 3D ego-motion based on Tracking fiducial markers

3.9 Extracting Lines with Hough Transforms

4. SGL Experimental Apparatus

4.1 Overview

4.2 Wireless measurement system

4.3 Processing of received wireless signal

4.4 Simultaneous CV and LTE outputs and processing

4.5 Upgraded Experimental system based on NI RF PXI

5. Discussion

Chapter 1   Overview of contract objective, phases and deliverables

1.1 Overview of the developed solution of SGL

The overall objective of the research activities comprising this contract is mainly that of the development of self geo-location (SGL) in which the mobile unit (denoted as the UE for user equipment) is to be located relative to the surrounding environment based on using LTE wireless signals of opportunity. To facilitate this we consider inputs as shown in Figure 1-1.

prior information and maps

UE

Signals from LTE and other wireless sources

GNSS signals when available

Camera sensor

Other sensors

- accelerometers, rate gyros, magnetometers
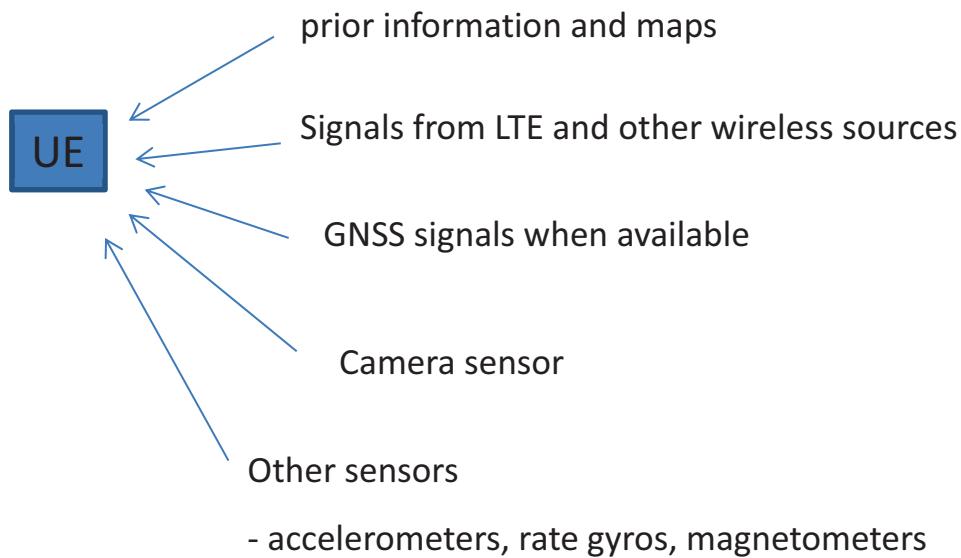
Figure 1-1 *Inputs to the UE for SGL*

Recent technology articles have outlined the race to have "Google Maps" extended to indoor locations, malls, airports, arenas and so forth. Smartphone manufactures such as Apple, Nokia, Samsung and RIM and ASIC manufactures such as Infineon, Qualcomm and Broadcom are scrambling to achieve processing that will provide sufficient resolution such that the detail of these available maps will be useful to the smartphone user. Such technology has obvious application for military based personal and asset location. Differences will be that the wireless signals of opportunity that can be used for such positioning are perhaps uncooperative with

poorly defined source locations and no guarantee of being accurately synchronized. The overall objective of the research is the enhancement of signal processing algorithms for estimating the location of a mobile station (MS) or user equipment (UE) based on wireless signalling. The wireless signalling is assumed to be sourced from 4G Long Term Integration (LTE) access nodes (AN) which are primarily used for mobile data and voice communications. The downlink signalling is specifically exploited, as the desire is for the UE processing to be capable of stand alone "self geo-location" (SGL). This implies that the network location processing or assistance is not involved. Further it is not assumed that the UE be registered with the LTE network but rather that it exploits LTE downlink signals of opportunity for the location estimate. Hence the relevant signalling will come from the LTE downlink reference signals (RS) that are synchronised across the LTE network of AN's. The signal structure of the RS's are assumed to be known by the UE as well as the location of the AN's.

Additional inputs can come from GNSS (GPS) signals when they are available. The camera input from a small web-cam like device integrated into the UE has been shown to be of very valuable input. Other sensors are the accelerometers, gyros and magnetometers which provide some information relevant to SGL. In this work development effort has been expended on the ancillary inputs provided by the camera and accelerometers.

The simplest scenario for the UE SGL within an LTE signalling environment is where the ANs are in known locations and synchronized in terms of the carrier frequency and code phase. This is typical of the CDMA IS2000 network that was studied in a previous contract in which the code phase synchronization tolerance is generally better than 50 nsec. The code offset of the pilot channel actually identified the particular base station, hence the code phase had to be synchronized to facilitate this. Also the IS2000 transmissions were calibrated in terms of delay such that they were within a 50 nsec tolerance relative to GPS time. This facilitated network based location services with the initial impetus being compliance with the e911 requirements. However, this is an added complexity that needs to be maintained that LTE dispenses with. Clearly in LOS propagation environments, the SGL tracking is robust provided that there are a

sufficient number of ANs available and that the geometry does not significantly compromise the GDOP as is illustrated in Figure 1-2. Here the UE is frequency and time synchronized with the set of AN's that are all located such that there is a LOS connection with the UE. Furthermore the AN locations are precisely known to the UE. It is assumed that the AN to UE ranges can be determined accurately. For the particular scenario depicted in Figure 1-2 there are only two unknowns, namely the (x, y) position of the UE and there are 4 range measurements possible from the synchronized AN's to the UE. Hence a single isolated position estimate can be achieved with redundancy. Not prior trajectory knowledge or assumptions of the UE is necessary.
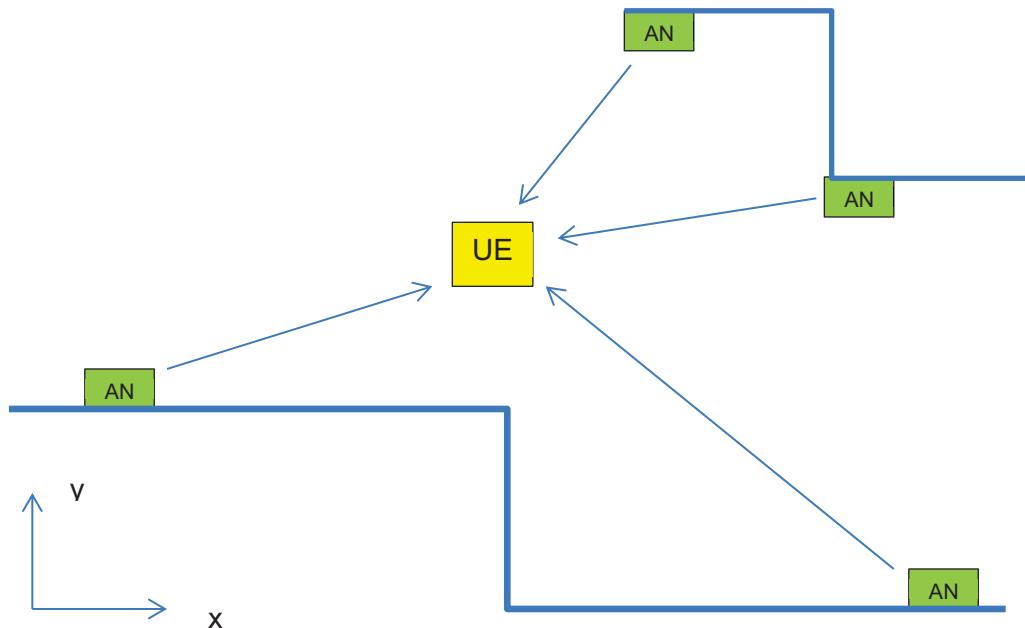
Figure 1-2 *Simplest case of SGL with all synchronized AN's in known locations and LOS propagation*

The implied assumption that the UE and AN clocks are synchronized is of course invalid as these clocks are independent with no direct physical or implied connection. Hence it is necessary to add the relative UE clock bias to the variable list such that we have (x, y, b) as the variable list. This type of SGL is equivalent to the GNSS navigation solution or the SGL within a terrestrial CDMA IS2000 network. Note that in the scenario depicted we still have a measurement redundancy allowing for an isolated position estimate of the UE.

Tight timing synchronization amongst the LTE AN's is not necessary for optimum data communication functions nor is it necessary for facilitating network based location, hence there is little impetus for the service provided to accurately time synchronize the network. Consequently, assuming that the bias of the code phase of the AN emissions is sufficiently accurately synchronized for SGL is not valid. Hence it is necessary to carry a state variable for each observable AN for the code phase bias. However, it is a reasonable assumption that the LTE AN signals are frequency synchronized as such reference signals are available from GNSS sources and back-haul links. As determined in phase I simulations, allowing for a static bias of the code phase of each AN did not significantly degrade the robustness of the SGL tracking even though there were more state variables to contend with. However, we now have more variables than observables for a single static location of the UE. A further complexity is that it is not guaranteed that the LTE AN is in a precisely known location which is likely the case in a non-cooperative scenario. This issue in addition to not knowing the code phase implies that isolated UE location estimates are not possible.

This significantly complicates SGL in that we have to generate additional measurements to exceed the number of unknowns. A key is that the AN's are assumed to be in static positions and are frequency locked. Consequently if two consecutive measurements of the AN-UE links are taken then we can amass more measurements than unknowns. However, the consecutive sets of AN-UE measurements are clearly linearly dependent and hence to not lead to a sufficient set of linearly independent measurements necessary for resolving all of the unknowns. However, if the AN is moved between the sets of UE-AN measurement sets then independent measurement sets result. This is illustrated in Figure 1-3.
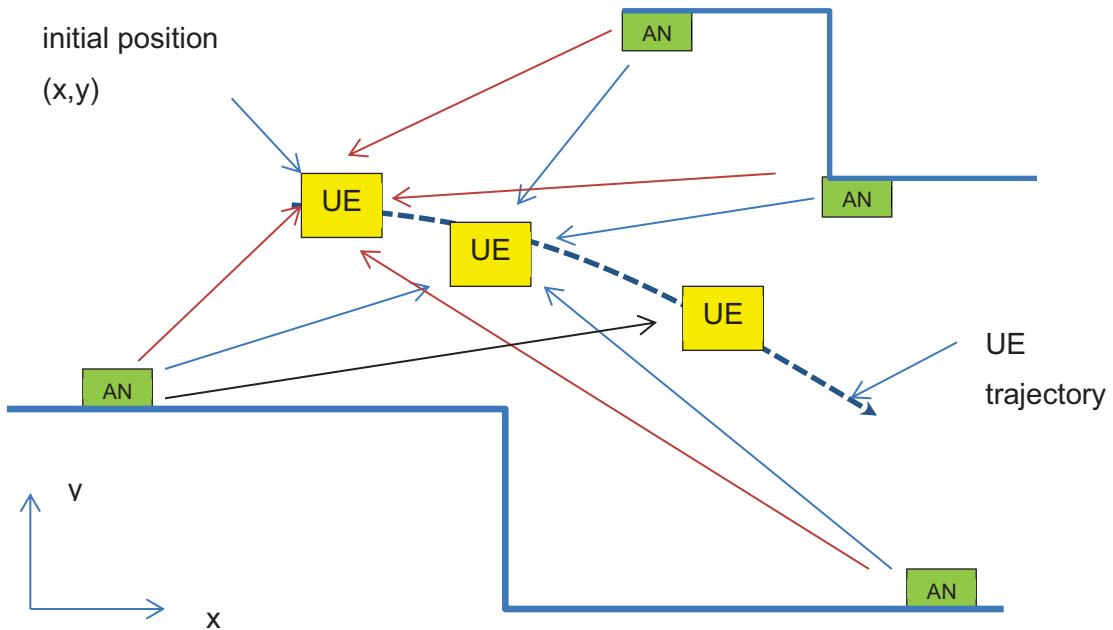
Figure 1-3 *Multiple sets of measurements generated by moving the UE*

Assume that a set of K UE positions are used and that for each position a complete set of measurements is available. If each measurement set is entirely independent then there is nothing gained from combining the K sets of measurements. However, the trajectory of the UE is typically quite smooth and also the random instability of the AN and UE clocks are highly correlated. Hence while we have not explicitly resolved the issue that the number of unknowns exceeds the number of measurements, then at least we have minimized the uncertainty of the variables. Consider it this way. If noise free measurements are available that exceed the number unknowns then deterministic calculations of the unknown UE position variables are possible. However, the measurements have a random component hence the estimate of the UE position will be random. With taking the K strongly correlated measurements into account where the UE is moved into K positions along a smooth trajectory, the issue is to exploit all of the correlations that are there as deterministic. The end result is that the moving UE just makes it more complicated to extract and exploit all of the underlying correlations.

Consequently we need a method that could efficiently use any prior information and disparate measurements that were related to the unknown variables in arbitrary ways. The only means

of robust SGL in such a context is to use Bayesian filtering methods. The Kalman filter (KF), extended Kalman filter (EKF) and the particle filter (PF) have been thoroughly investigated in this context as part of this research. The complexity of the problem revealed that additional tools were needed to accommodate the numerous and disparate constraints and observations. This was really the impetus for involving the Simultaneous Location and Mapping (SLAM) algorithms that were developed for the LTE SGL. It was discovered that in many scenarios of practical interest that the UE could use signals of opportunity from the LTE sources. That is, just knowing that the LTE AN transmission comes from a stationary source is sufficient to gainfully use it in a SLAM context. Even if all of the AN's are in unknown locations then, in principle, it is possible to simultaneously locate the AN's and the trajectory of the UE. Also a more efficient fastSLAM method was developed for the SGL. An overview of these Bayesian methods is given in chapter 2 with the details presented in part II of this report.

To proceed with a Bayesian approach we make the justifiable assumption that the AN's are assumed to be stationary and that the AN signals are frequency locked. Note that this does not imply that they are time synchronized such that the code phase is arbitrary but the offsets are relatively static in comparison to the AN emissions. A separate bias for each AN emission is then assumed and denoted by the variables $(b_1,...b_M)$ which are unknown but constant with time. In Figure 1-3, the UE is moved while taking range measurements with the underlying assumption that a Bayesian estimator can be realized that incorporates the assumption of a smooth trajectory with an uncertainty imposed at each time step. The Bayesian filter can also take into account the prior knowledge of the UE position from the last known position or user input relative to a supplied map.

A significant reduction in the uncertainty of each of the UE update steps is possible if computer vision (CV) inputs are used from a small webcam that can easily be integrated into the handset device. As described in chapter 3, CV provides an accurate measurement of short trajectory lengths on the order of several tens of meters. Figure 1-4 outlines the basic concept of using CV ego-motion relative to a ground surface for generating updates for the Bayesian filter. Even if these inputs drift over longer trajectory lengths they significantly reduce the update uncertainty that the Bayesian filter has to otherwise contend with. Note that the CV measurements add tighter constraints between the UE positions by removing much of the uncertainty of the UE update.
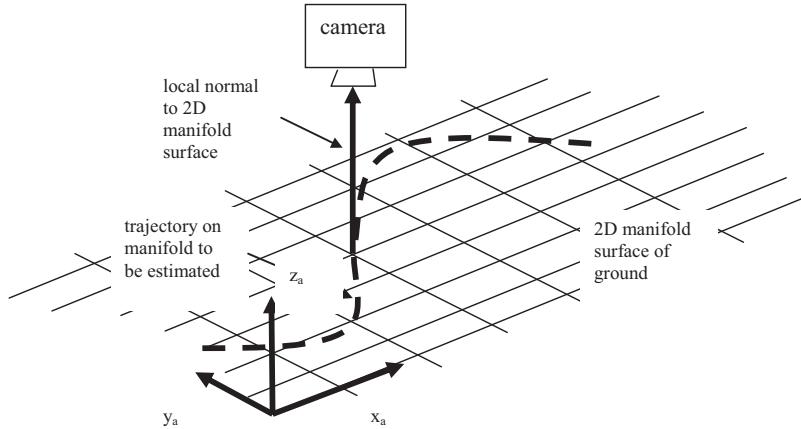
Figure 1-4 *Ego-motion of camera sensor relative to a 2D manifold surface*

A challenge is that the CV problem is not trivial as the UE is hand held and hence is subject to random motion in 6 DOF (3 translation and 3 orientation angle variables). If it can be assumed that a flat levelled floor is being observed then the two tilt angles of the camera can be measured absolutely. As such only the azimuth angle and the height of the camera needs to be added as nuisance variables to the state variable list. To robustly measure these tilt angles and hence provide a direct CV based trajectory ego-motion output that can be used in the Bayesian estimator, a Kinect type of projector/camera combination is considered. A stereoscopic camera was also considered and partially developed for this purpose. These means were demonstrated to work extremely well however, the additional hardware complexity is a factor. As well it does not fit the handheld form factor. Hence monocular webcam solutions are currently being considered as no additional hardware is required.

Additionally 3D CV perspective transformations have been developed in phase II as providing a potential solution to resolve the unknown nuisance variables associated with the UE/camera trajectory. A promising possibility for absolute trajectory start point referencing is that of the CV recognizing a wall mounted icon, template or other geometric structure. The perspective transformation is used in this case as is illustrated in Figure 1-5. The 2D template is shown on the right side with the perspective on the left side. From the vantage point of the camera the perspective transformation can be determined from which the 6 DOF variables of the relative camera position can be determined.
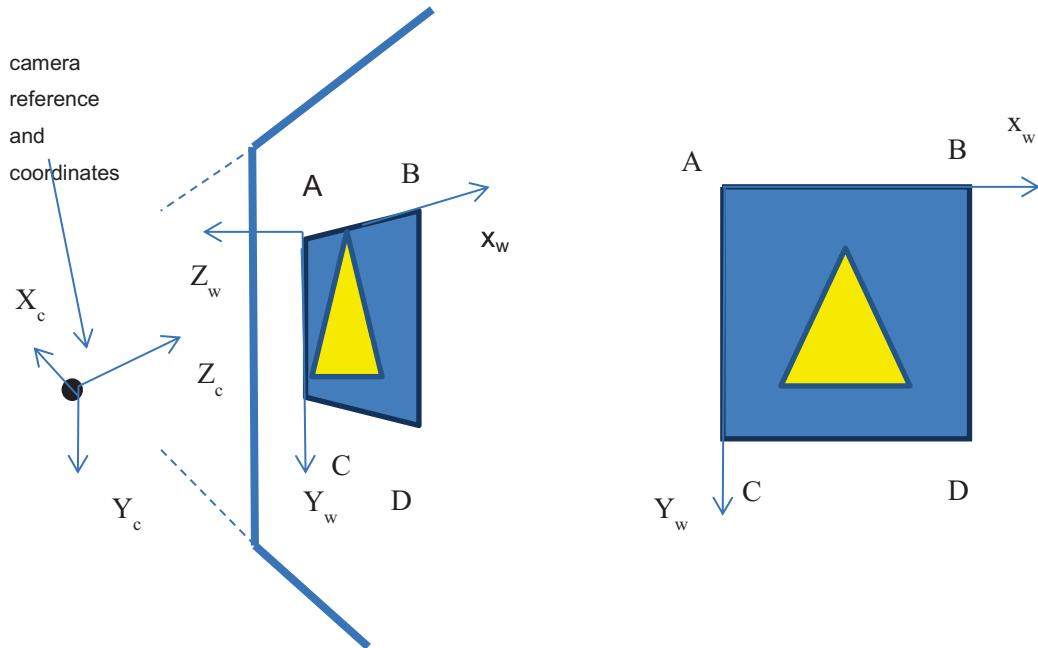
Figure 1-5 *Inverse perspective transformation obtained of a known template on the wall*

CV processing with the standard web camera, which is compatible with a handset device, is sufficiently capable of estimating this perspective transformation. Naturally, if such templates or icons are readily observable throughout the UE trajectory then one can question the requirement for the LTE signalling. However, icons are not always available, in known locations or sufficiently distributed that they are visible throughout the UE trajectory. Certainly in airport lobbies or commercial buildings such icon systems will inevitably be implemented. However, in more hostile or unknown territory scenarios, the existence of usable icons is not guaranteed but is probable. Hence the development has been extended to markers of opportunity as observed in the FOV of the camera. Also as will become evident is that the CV observables are useful in obtaining accurate assessment of short trajectory lengths but that ego-motion based on CV alone drifts. Hence the combination of CV and LTE wireless observables are complementary.

A major limitation with SGL based on LTE signals is the multipath propagation effects which typically randomly disperses the signal such that delay spread can be up to several hundred nsec. Hence the deviation of the range measurements can be several 10's of meters due to multipath alone. This renders the LTE signals of very little information content in the SGL context. Certainly the wide 20 MHz bandwidth helps as the resolvable components of the

multipath can be eliminated leaving only the initial group of multipath components corresponding to the leading edge and hence reducing the range uncertainty to about 30 meters. Also there is progress in LTE technology development to push the bandwidth up to 100 MHz. However, with current bandwidths, the delay spread in NLOS environments is such that the uncertainty in the pseudo-range measurement is too large to be of practical use for indoor SGL.

In phase II consideration was given to representing the multipath delay bias with a random process as the bias will change between LOS and NLOS conditions. The alternative is a Boolean switch that indicates states of high LOS and high NLOS. The detector processing for such a switch is based on the synthetic array. Note that the AN-UE propagation path will be LOS in certain locations. The SA can be used to accurately access the spatial coherency of the incident LTE wireless signal as illustrated in Figure 1-6. Here the spatial coherency of the AN signal is evaluated as the UE is moved. If the spatial coherence metric, denoted by γ_{LOS}, is high then a LOS condition is declared.
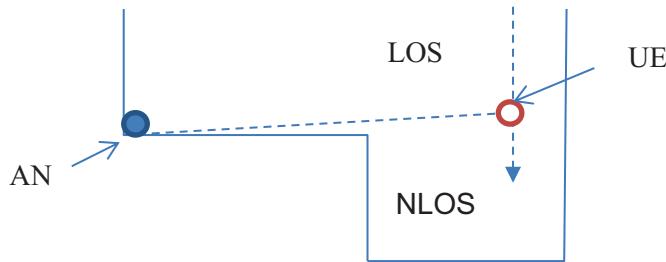
Figure 1-6 *Illustration of the boundary between the NLOS and the LOS region.*

Based on this assessment various options are available.
1. Trigger a direct calculation of the estimate of the AN location relative to the UE trajectory if γ_{LOS} is sufficiently high.

2. If a radio map exists that is based on γ_{LOS} then modify the belief map of the location of the UE in a Bayesian sense that takes the overall UE trajectory into account.

3. It is necessary to use a state variable in the SGL SLAM that is related to the delay bias between the UE and the AN. Different values of the delay bias state variable can be used

depending on the condition of multipath. Hence γ_{LOS} can be used to select the appropriate model of the delay bias for the currently encountered propagation condition.

4. Multiple pseudo range measurements are made at each location of the UE as it moves along the trajectory. The quality of these measurement in terms of the SGL objective is generally higher for LOS vs NLOS conditions. Hence γ_{LOS} can be used to form a weighting factor related to how reliable a given UE-AN pseudo-range measurement is.

5. RSS measurements are easier to generate by the UE and can also be incorporated as a set of measurements for the SGL. If γ_{LOS} is low then a predominantly NLOS condition exists in which the RSS observables often provide more information than the pseudo-range measurements. Hence the weighting of point 4 can be extended to the RSS observables.

Figure 1-7 illustrates the trajectory of the UE intercepting the spherical wave emanating from the AN. If the local trajectory can be accurately determined say from a CV input then if γ_{LOS} is such that there is a LOS condition then it is possible to determine the location of the AN relative to the trajectory as stated resulting in a high information observable.
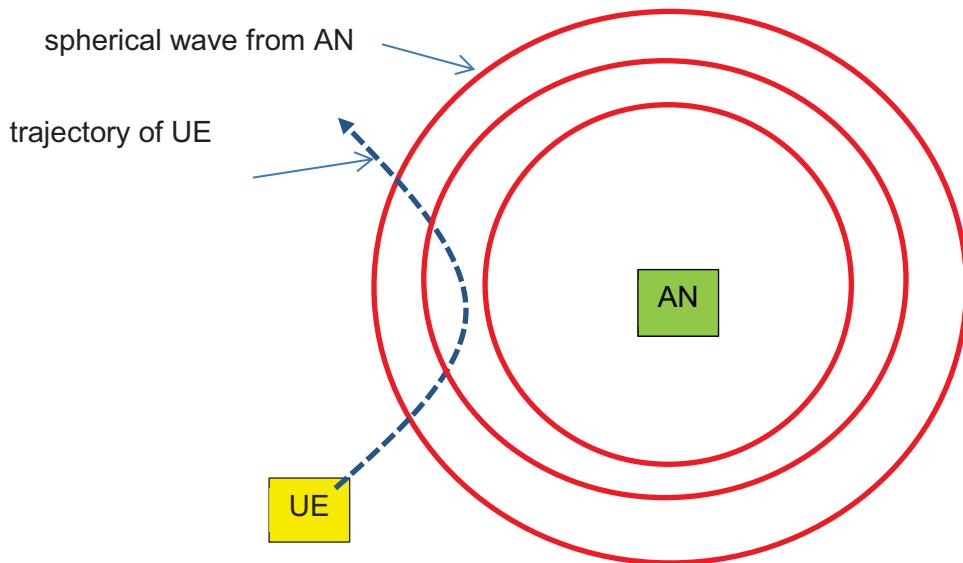
Figure 1-7 *Assessment of the spatial coherence by continuous sampling of the AN complex gain*

An overall flow chart of the SGL processing is given in the Figure 1-8. Initially prior maps and information are used by an initialization phase which configures the SLAM algorithm for SGL. The initial displacement increment is taken along the trajectory and the observables are sampled.

```
prior maps          ──►  initialization phase
information

                    ──►  sample observables

                              SLAM update  ──►  UE trajectory and
                                            ◄──  position output
                                                 outlier detect
                         check for LOS conditions   recalculate as necessary

      invoke the synthetic array processing          state variable
                                                      storage
```
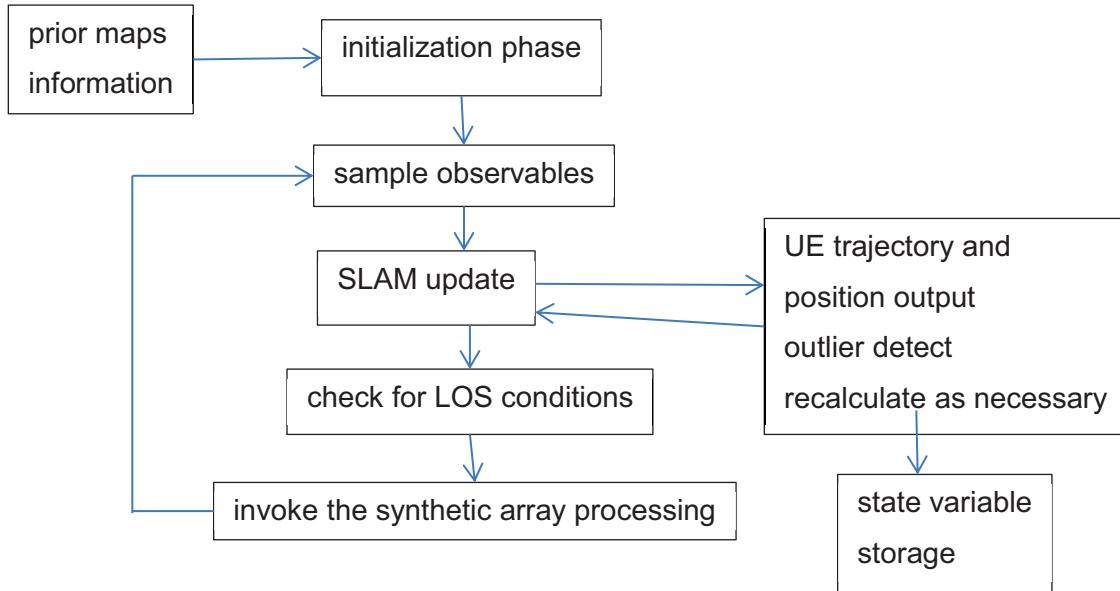
Figure 1-8 *Overall flow of SLAM algorithm*

Initialization phase – the UE is enabled for the SGL operation. Any prior maps are read into the device along with knowledge of where it was last time it did an SGL estimate. Any user input information is of course also applied. Also radio maps are loaded in corresponding to the approximate location. These will include the list of possible AN's in the vicinity that are known. Additionally, the statistical trajectory information will be input and processed. This may include knowledge of whether the UE is mounted to push cart, vehicle, foot soldier or other asset. A search of GPS and perhaps other GNSS sources is done with a listing of the available sources for clock synchronization and possibly AN sources. Next the UE clock offset is estimated from the set of observable signals. It is assumed here that the UE is relatively stationary and that the signals are categorized for their quality in terms of use as frequency reference signals. A model for the clock oscillator instability is then determined as will be used in the SLAM method. Then the UE camera is used to scan for a LandMark (LM) that can correspond with the map. The user can optionally assist in this stage by scanning the camera. Of course if there are no relevant

maps available then this step is skipped. Based on the observed CV LM's and the set of GNSS signals and AN signals that are available the SLAM algorithm is configured along with an initial belief map of the UE (x,y) (and possibly z) position that is used for the SLAM.
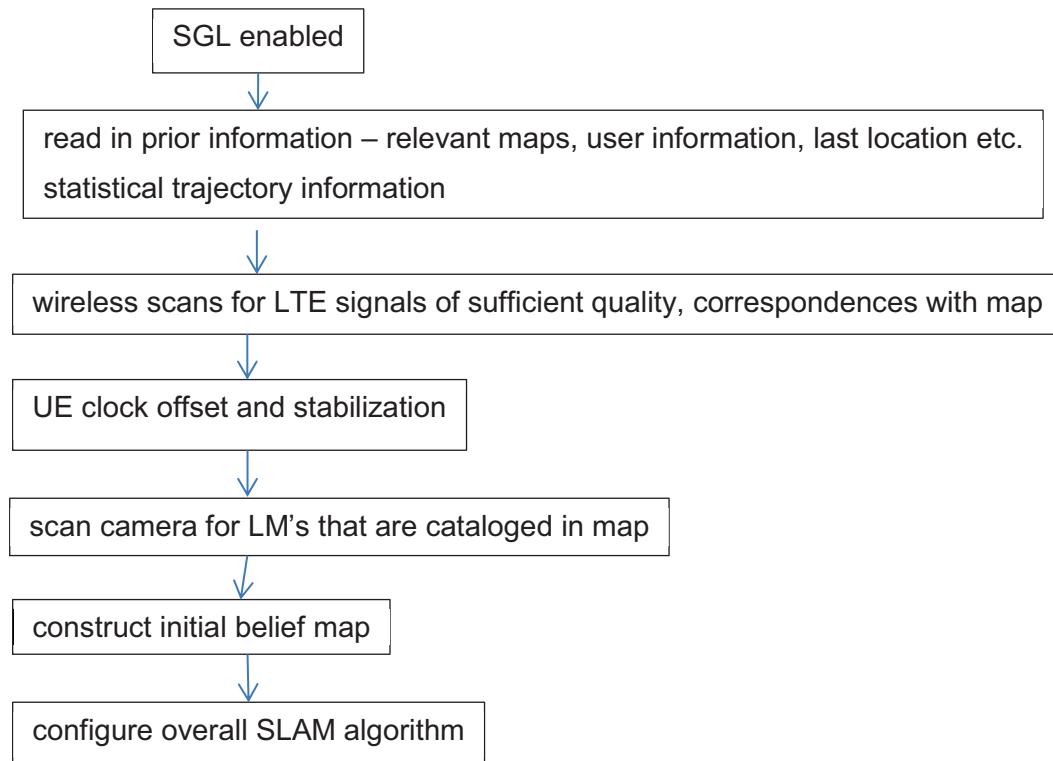

Figure 1-9 *Initialization phase*

Sample observables

The various observables are sampled as configured from the set in the initialization step. These involve the wireless signals from the LTE, WiFi, RFID or other sources of opportunity, any available GNSS signals, camera inputs as well as inputs from MEMS inertial sensors and magnetometers.

SLAM update

The fastSLAM method is used for the Bayesian filter for updating the trajectory and the feature point information. The most efficient combination is the PF used for the UE displacement and the EKF for tracking nuisance variables and converging map FP's.

```
┌──────────────────────────────────────────────┐
│ trajectory update model                        │
│   - update particles representing position     │
│   - EKF representing mapping and nuisance variables │
└──────────────────────────────────────────────┘
              │
              ▼
┌──────────────────────────────────────────────┐
│ Determine the EKF  Taylor components           │
│ based on the observations and update all of the EKF's │
└──────────────────────────────────────────────┘
              │
              ▼
┌──────────────────────────────────────────────┐
│ determine the particle weights and resample the particles │
└──────────────────────────────────────────────┘
```
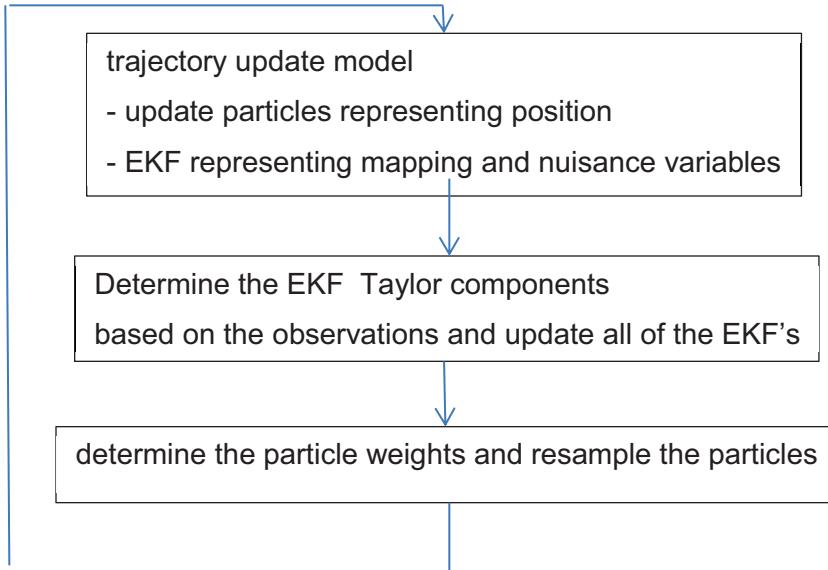
Figure 1-10 *SLAM update*

Finally, the possibility of a map is being considered for the SGL activity. Such a map, if available, would provide the UE with an outline of the building from which a prior belief map of location can be assessed. It would show the position of permanent obstacles and walls such that probabilities of estimated trajectories can be assessed. This is a very useful input for the various Bayesian filters that are being developed. The map could also provide location of templates that can be observed by CV and perhaps the locations of AN's. In hostile scenarios where a partial map is available at best, an outcome of the SLAM could be to slowly anneal such a map and add features to it. This is a powerful tool especially if there are several UE's in the area cooperatively building up such a map.

1.2 Outline of the report

The organization of the remainder of this final report is as follows:

Chapter 2. Overview of the SLAM algorithm for SGL – This chapter describes the SLAM algorithm as applicable to the SGL problem. The general set of observables from the AN wireless downlink transmissions and the CV observables will be described in the context of being inputs for the SLAM algorithm. The PF from the phase I will be reviewed followed by a development of the SLAM based on the Kalman filter (KF), extended Kalman filter (EKF) and then the fastSLAM which involves a combination of the EKF and PF. fastSLAM will be what is used specifically for the SGL algorithm development.   The details of the SLAM algorithm research are contained in Part B. Also in this section is a detailed description of the downlink LTE signal. As described, the resource blocks in the reference channel contain the coherent synchronization blocks from which the signal phasing at the mobile can be estimated.

Chapter 3. CV Ego-motion - There have been many new developments in the CV ego-motion that are of significance to the objectives of this contract. These have been developed to the point that integration with the SLAM algorithm is practical. The CV forms an independent set of observables that is useful for short term trajectory estimation.   Not only is the CV ego-motion used as SLAM observables but also as a means of testing the algorithms for the LOS AN wireless beamforming. Given a good set of feature points, the ego-motion is sufficiently accurate to precisely track the translation and orientation of the wireless receiver.

Chapter 4: Development of Experimental Hardware – Several iterations of the hardware have been undertaken with the current unit, capable of some simultaneous recording and fusion of wireless and CV signals in the SLAM context. The receiver processing consists of two parts. The first is the signal is down-converted, sampled and stored. Then post-processing simulates the SLAM tracking algorithms along with the synthetic beam forming.

Chapter 5: Discussion - Overall observations and outcomes of the theoretical and experimental outcomes and suggested next steps.

Chapter 2 SLAM SGL Algorithm

2.1 Overview

In this chapter the outline of the SLAM method developed for the SGL system will be described. The bulk of the details are in part B of the report which describes the OWLS (Opportunistic wireless location system). As described in Chapter 1, the SGL problem involves the estimation of the UE trajectory with LTE wireless signals emanating from AN's from which range measurements are extractable. AN's occur in two types: Anchor nodes where the location of the AN is known which is denoted as ANan and feature nodes of initially unknown position denoted as ANfn. In either case there are additional unknowns such as the delay bias attributed to each UE-AN link as well as the orientation of the UE which affects the antenna response. SLAM processing is ideally suited for this type of SGL problem and will be developed in this chapter.

The basic assumptions are as follows:

1. LTE wireless signalling will be used for the observables

2. Only downlink LTE signals will be assumed. The UE is not registered with the network and hence uses the LTE downlink signals as signals of opportunity.

3. The LTE downlink signals from the various AN's may or may not be accurately time synchronized. Both modes will be considered.

4. The UE has to have all the information regarding the SGL estimate such that a handset based location solution is possible for an uncooperative network.

5. Multipath will be variable as indoor and dense development areas are of primary interest.

6. The UE is expected to be undergoing some arbitrary motion that varies in terms of smoothness.

7. UE is not frequency synchronized to the AN in any way. That is it must be assumed that the UE has a free running clock.

8. CV inputs are available for updates of the UE pose.

9. SGL algorithm must accommodate the possibility that the AN locations are not accurately known.

The overall objective is to generate a belief map of the location of the UE as the trajectory unfolds. The belief map will provide a posterior estimate of the location of the UE that is iterated every time interval which is based on all available data up to the current time. For such estimation problems based on the aforementioned assumptions, it is generally not valid to assume Gaussian statistics. Consequently Kalman filter (KF) type parametric Bayesian filters (BF) are typically sub-optimal. The particle filter (PF) has been developed for the SGL problem as described in the first interim report. However, a limitation is that the number of state variables is large such that a mix of EKF and PF is necessary which is the fastSLAM algorithm that will be developed for the SGL problem.

Indoor environments are subject to severe multipath which significantly distorts wireless signals in the LTE frequency bands. Even though LTE has a potentially large 20 MHz bandwidth, this is typically not sufficient to provide adequate accuracy of time of flight measurements for indoor SGL. Multipath distortions it itself renders traditional isolated location estimate methods of SGL virtually useless for indoor applications where an accuracy of 1 meter is required. Combatting the effects of multipath is the most significant challenge in providing adequately accurate LTE based SGL. This is exasperated by the fact that the AN's are not necessarily synchronized nor in accurately known positions. This implies that there is a random delay offset that must be considered with regards to each AN that cannot be considered to be a Gaussian random variable. The only way to mitigate these limitations is if the relatively smooth trajectory of the UE is exploited by the BF. With a smooth trajectory, multiple measurements between the UE and the same AN can be combined. The other key point is that the AN is assumed to be stationary. Additionally it is optionally assumed that CV inputs exist such that the relative short-term update motion in terms of translation and orientation can be determined and passed to the BF as statistical UE update information. The CV will be assumed to consist of a small webcam which is compatible with the monocular 3D vision systems. An overall imposed constraint is that the UE needs to be self-contained and that the apparatus and

processing required for the SGL function must be commensurate with the handheld form factor hence the monocular limitation.

Initially pure LOS links will be assumed between the UE and the various AN's. However, these will be supplanted with NLOS multipath links which will be accounted for by assuming a time of flight (TOF) bias that varies with time. Hence it is not really a bias but more a slowly varying random variable representing the bias. This bias can switch quickly when moving from LOS to NLOS conditions. As described previously, Boolean switch models and Markov models have been analysed for the fluctuating delay bias issue. We have also analysed the possibility of generating a metric based on the equivalent of the Ricean K factor which provides an approximate measure of the level of multipath in the LTE wireless signal sources.

 Boolean multipath model switches are compatible with the Bayesian filter however with the expense of carrying an additional state variable. An example of a scenario where the propagation changes from LOS to NLOS is shown in Figure 2.1. The concept would be that the processing would have to recognize that there has been a change in the propagation perhaps from the coherency of the carrier phase as the UE propagates. This is where the CV updates will have a significant impact as they will provide a short term trajectory that is sufficiently accurate to facilitate beamforming on the AN signal to determine it's spatial coherency. Hence the quality of the LOS component can be determined and different models can be switched into the SGL SLAM algorithm.
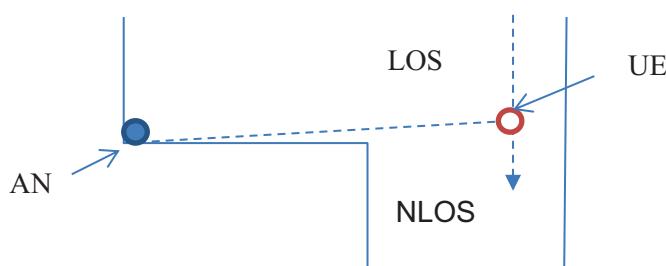
Figure 2.1 *Illustration of the boundary between the NLOS and the LOS region. In theory, as the UE moved across this boundary, the bias model would switch from LOS to NLOS with appropriately selected delays*

The LOS/NLOS state variable can take several forms. The first is a Boolean indicator whether the signal carrier phase is sufficiently coherent such that a LOS threshold is surpassed. Hence the Bayesisan filter can incorporate both the LOS and NLOS propagation models for the signal. For the LOS case a fixed excess propagation delay can be assumed. In the pure NLOS case, any time delay observables are of little value to the overall SLAM and the RSS observables are used instead. While individual RSS measurements are low in information value, the sequence of RSS measurements can give approximate location information where TOA based observables are useless.

Another implementation is that the excess propagation delay is a state variable. The receiver demodulates the resource blocks of the LTE signal which are narrow bandwidth coded signal bursts that are then Fourier transformed to obtain the channel time domain impulse response. The TOA estimate of a given LTE signal is then taken as the time of the leading edge of the transformed impulse response offset by the parameter "b" which is determined by the PF. The higher the aggregate bandwidth of the LTE downlink signal the sharper the leading edge transition of the estimated channel impulse response will be which will result in less variation of "b". In turn, this implies that the spread of the posterior pdf of the state variable belief is more confined resulting in an overall lower position variance.

In summary, the key to indoor positioning based primarily on wireless signalling is that the there is a sequence of correlated measurements. That is the trajectory of the UE is relatively smooth. This correlates the observables such that even if the trajectory is random, a steady state covariance of the state variable vector is achieved.

The Bayesian Fisher information matrix (BFIM) is derived as a central key to the SLAM-based opportunistic wireless localization system (OWLS) that unifies all information from observables, the MN trajectory, and a priori knowledge in one single matrix. The BFIM illustrates that observability of a SLAM-based OWLS is achieved under the assumptions of stationary ANs and

the MN smooth trajectory. The development of OWLS is convoluted and detailed in places. Consequently, in an attempt at a coherent overview, the details of OWLS are deferred to part B of this report.

What makes indoor positioning difficult in the LTE SGL scenario is that there are so many unknowns. If the UE trajectory was the only unknown with the AN's synchronized and in known locations, and the propagation LOS then the SGL would be that of the conventional GPS navigation problem which is very well understood and generally robust. Unfortunately, in the LTE SGL context, the location of the AN's is generally unknown as is the code phase of the signal. Furthermore the multipath propagation results in a TOA bias that is unknown and is not stationary. Further the clock of the UE is not synchronized to any reference point in the network. Hence SGL is a problem where the UE trajectory is a small component of the overall list of unknowns with the bulk of the unknowns being the vast list of nuisance parameters involving the AN's and the multipath propagation. For this reason the development of SLAM (Simultaneous Localization and Mapping) has been pursued. SLAM is a promising algorithm for this situation as the mapping component can be more abstract than merely determining the position of the AN's as part of the map discovery. SLAM will simultaneously determine the unknown parameters of the AN's and the trajectory of the UE itself. Hence the "map" component consists of the long list of nuisance parameters. The key result of the early work in SLAM was that there is a high degree of correlation between estimates of the location of different landmarks in a map and these correlations would increase with future observations [1]. Later in decade of 1990, the key research on convergence were developed by Csorba [8] and then on KF based SLAM methods and the probabilistic localisation and mapping methods by Thrun [9]. The conceptual breakthrough in SLAM research formulated around this time was that it was generally convergent. It was recognized that the correlations between landmarks were the critical component that enabled convergence.

To describe SLAM in the SGL context begin with a consideration of the isolated UE for which we wish to estimate the position. Clearly this is impossible without any observations so assume an AN that generates a recognizable pilot signal as observed by the UE which perhaps does a TOA measurement of the signal relative to it's own clock. Unfortunately this provides no information as the AN location is not known, the downlink signal is not synchronized and the phase of the UE clock is random. Now suppose that the UE moves and at an other time instant observes the AN signal. As the two UE positions are assumed to be independent, nothing is gained from the second observation. If the UE was stationary such that the measurements were totally correlated, we would just have two measurements of the same thing. In fact the measurements would still be uncorrelated as the clocks are not synchronized and that the time interval between measurements is assumed to be entirely random.

But now suppose that the time between measurements is a fixed time set by the UE and that the frequency offset and drift between the clocks has been noted by the UE. Then based on the sequence of two measurements it is possible to say that the UE radial distance from the AN has changed by an estimated amount subject to some uncertainty traceable to the relative instability of the clocks and perhaps TOF variations due to multipath. However, this is information regarding the UE trajectory albeit modest. Now suppose that the UE is moved by an operator who is constrained to some form of stochastic model. That is we know the operator is walking and not in a high speed vehicle for example. Now the radial position information can be combined with the user motion dynamics model to present a refined estimate regarding the trajectory. Now consider this as extended over a number of observations as the UE is moved. We can infer plausible trajectories from the sequence of observations and the prior assumptions of the dynamics model. The end result after the sequence of observations is a belief map or PDF of the possible UE trajectory relative to the unknown position of the AN. Even though the absolute position is not known, there is quantifiable information in the resulting conditional belief map.

Next consider another AN from which the UE can make observations. The only assumption is that both AN's are stationary. The UE is moved while a sequence of observations is made. Without observations, as time progresses, the PDF of the UE location will diffuse due to uncertainties in the UE motion itself. The observations themselves are subject to clock instability and variations in the multipath delays. However, the net information of the observations will tend to offset the probability diffusion that occurs due to the UE dynamics itself. It is the correlation between the sequence of measurements that provides the information rather than the isolated measurements themselves. SLAM will extract all of the information contained in these correlations and establish a joint belief map of all of the unknown variables. These include not only the UE trajectory but also the nuisance or mapping variables.
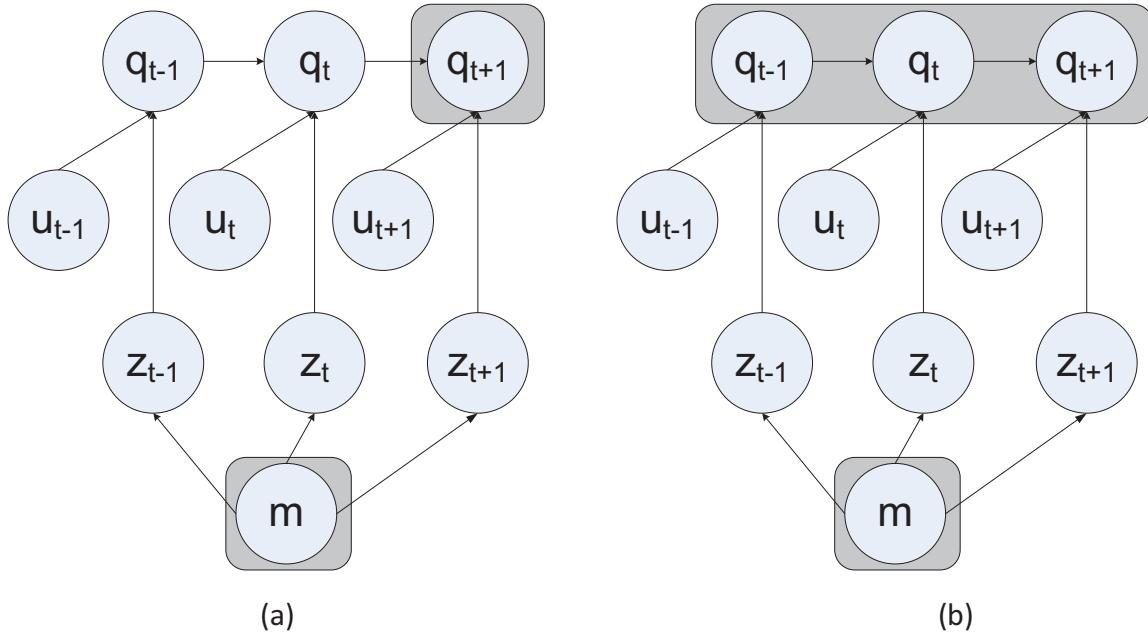
Figure 2.2 Graphical schemes of a) online SLAM in comparison with b) full SLAM

As the UE is moved and the location of the M fp's and the UE trajectory is estimated, the errors in the state variables are correlated [2]. This correlation is exploited in SLAM and can be conceptualized as an emerging mesh of constraints as illustrated in Figure 2.2. Here z denotes the measurement, m the variables associated with the AN's and other nuisance parameters. u

denotes the UE position or the trajectory and q the dynamics model of the UE location update. Typically an on-line SLAM algorithm is required that continuously processes the information and provides the output of the most current estimate of the UE trajectory and mapping variables. This is illustrated in Figure 2.2a. In Figure 2.2b the off-line SLAM is illustrated in which the overall UE trajectory estimate and mapping variable estimate is done off line. Hence the overall trajectory is not available until all of the observations have been made. While the off-line gives better performance as it uses all of the measurements made it has an obvious latency issue such that it is not suitable for real time applications. One illustration of the SLAM constraints is that of representing the individual constraints as springs between the fp's and UE pose as shown in Figure 2-2. An observation at each time is analogous with a displacement in spring network and it correlation effects on neighbors depends on their distances to other feature points, i.e. , the nearer they are, the greater is effect. As the UE moves and received more observations of feature points, the correlation between feature points' estimates increase and the spring become stiffer. Noise and un-modelled uncertainty can be imagined as disturbances to this spring model that distorts the estimates of the state variables. Hence the stiffer the springs, the more robust the solution and the less the influence of these noise disturbances will cause errors. Note that the map features could be in known locations. As such they are anchors from which absolute location reference can be inferred. Also map features could have various degrees of freedom. For instance, a floor marker may not be unique but may be oriented in a specific direction. A difficultly that is encountered with features in SLAM is that of establishing correspondence. For active AN's, this is not an issue as the signal is generally coded such that it can be uniquely identified relative to the other AN signals. However, if the AN coding is not known beforehand then it is difficult to separate it from the other AN emissions compounding the correspondence problem. SLAM maps for SGL can extend beyond the AN's as CV is used. Hence while CV is powerful in terms of establishing and using map fp's there is the issue of determining correspondence as will be developed in the following chapter.

2.2 SLAM Information Matrix

SLAM in the SGL context is a unifying method that can accommodate multiple disparate observations over multiple time intervals. It can also accommodate what is known and unknown about the various features provided that the uncertainty can be quantified by reducing it to a set of covariances. SLAM thereby forms a unified framework for fusing the information from all of the disparate bits of information, constraints, measurements and so forth into a single information matrix. It is therefore ideal for the LTE/CV/SGL problem formulation.

   The concept of the information matrix, denoted in general by $\mathbf{J}$, is key to the SLAM algorithm. It can loosely be regarded as a quantitative summary of all of the information regarding the unknowns in terms of the map variables and the UE trajectory in one matrix. The relation to quantifiable information is only valid for the case where the problem is linear and jointly Gaussian however for problems where the nonlinearity is mild and that the measurement and update noise is unimodal and approximately Gaussian then the information matrix has deep significance. To develop the information matrix assume that the MLE of a set of variables of q can be approximated as being jointly Gaussian represented by $N\left(y, \mathbf{J}^{-1}\right)$. Here q represents the actual value of the state variable vector expressed as

$$q = \begin{bmatrix} q_1 \\ \vdots \\ q_t \\ m \end{bmatrix} \tag{1}$$

For typical problems that are not strictly Gaussian, the approximation of the PDF of the MLE as $N\left(y, \mathbf{J}^{-1}\right)$ is still generally valid. Regardless, $\mathbf{J}$ is the information matrix and is the inverse of the covariance matrix of the MLE estimators for q. The concept of the information matrix is best exemplified through the following set of simple examples.

Consider a case where the UE undergoes random motion in the x direction according to a first order markov (random walk) model. Consider the initial step where $x_0 \sim N\left(0, \sigma_0^2\right)$ which is of course the prior PDF. Now consider the step between x_0 and x_1 as shown in Figure 2-4.
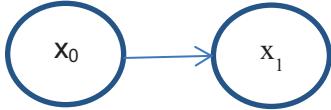
Figure 2-4 *One step problem*

The update from t=0 to t=1 is characterized by

$$x_1 = x_0 + u$$
$$u \sim N\left(0, \sigma_u^2\right) \tag{2}$$

The joint posterior PDF of $\{x_0, x_1\}$ is given as a normal of zero mean and covariance of

$$Q = \begin{bmatrix} \sigma_0^2 & \sigma_0^2 \\ \sigma_0^2 & \sigma_0^2 + \sigma_u^2 \end{bmatrix} \tag{3}$$

Note that the covariance of x_1 given x_0 is denoted as $\sigma_{1|0}^2$ is determined as

$$\sigma_{1|0}^2 = \left(\sigma_0^2 + \sigma_u^2\right) - \frac{\sigma_0^4}{\sigma_0^2} = \sigma_u^2 \tag{4}$$

which makes sense. Likewise the covariance of $x_0 \mid x_1$ is

$$\sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_u^2} = \frac{\sigma_0^2 \sigma_u^2}{\sigma_0^2 + \sigma_u^2} \tag{5}$$

which for the case where $\sigma_0^2 \gg \sigma_u^2$ is about σ_u^2. If there is no measurement then the variance of the estimate of x_1 is $\sigma_0^2 + \sigma_u^2$ which is also consistent with the covariance matrix. The inverse of Q is given as

$$J = Q^{-1} = \begin{bmatrix} \dfrac{\sigma_0^2 + \sigma_u^2}{\sigma_0^2 \sigma_u^2} & -\sigma_u^{-2} \\ -\sigma_u^{-2} & \sigma_u^{-2} \end{bmatrix} \tag{6}$$

Note that J can be expressed as

$$J = \begin{bmatrix} \sigma_u^{-2} + \sigma_0^{-2} & -\sigma_u^{-2} \\ -\sigma_u^{-2} & \sigma_u^{-2} \end{bmatrix} = \begin{bmatrix} \sigma_0^{-2} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \sigma_u^{-2} & -\sigma_u^{-2} \\ -\sigma_u^{-2} & \sigma_u^{-2} \end{bmatrix} \tag{7}$$

which is the combined information from the initial condition and from the update. Note that

$$\begin{aligned} [J]_{11} &= \sigma_{x_0|x_1}^{-2} \\ [J]_{22} &= \sigma_{x_1|x_0}^{-2} \end{aligned} \tag{8}$$

Consider another problem where we have a measurement associated with an anchor point of m_1 as illustrated in Figure 5. Assume that the variance of the measurement of the anchor point is σ_z^2.
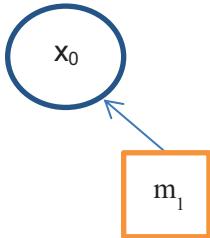
Figure 2-5 *Single measurement of an initial trajectory point*

The covariance of x_0 given the measurement is

$$\sigma_{0|z}^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_z^2 + \sigma_0^2} = \frac{\sigma_z^2 \sigma_0^2}{\sigma_z^2 + \sigma_0^2} \tag{9}$$

Now suppose that we consider the measurement as an amendment to the prior one step problem as shown in Figure 2-6. The two steps with the measurement of the anchor feature. From the graph we can infer

$$\begin{aligned} \sigma_{1|0,z}^2 &= \sigma_{0|z}^2 + \sigma_u^2 \\ &= \frac{\sigma_z^2 \sigma_0^2}{\sigma_z^2 + \sigma_0^2} + \sigma_u^2 \end{aligned} \tag{10}$$
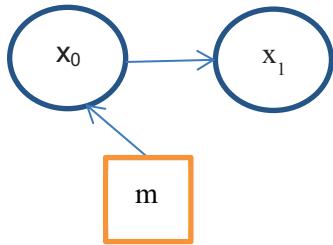
Figure 2-6 *Single step with a single measurement*

Alternately, consider the state vector of $q = \begin{bmatrix} x_0 & x_1 \end{bmatrix}^T$:

1. first start with the covariance matrix of the positions

$$Q = \begin{bmatrix} \sigma_0^2 & \sigma_0^2 \\ \sigma_0^2 & \sigma_0^2 + \sigma_u^2 \end{bmatrix} \tag{11}$$

2. The information from the measurement is given as

$$J_z = \begin{bmatrix} \sigma_z^{-2} & 0 \\ 0 & 0 \end{bmatrix} \tag{12}$$

as it refers to the x_0 position state.

3. Compute the total information

$$J_{tot} = Q^{-1} + \begin{bmatrix} \sigma_z^{-2} & 0 \\ 0 & 0 \end{bmatrix} \tag{13}$$

4. Determine the inverse

$$Q_{tot} = J_{tot}^{-1} \tag{14}$$

For the current example we can determine that

$$Q_{tot} = \frac{\sigma_0^2 \sigma_z^2}{\sigma_0^2 + \sigma_z^2} \begin{bmatrix} 1 & 1 \\ 1 & \frac{\sigma_0^2 \sigma_z^2 + \sigma_z^2 \sigma_u^2 + \sigma_0^2 \sigma_u^2}{\sigma_0^2 \sigma_z^2} \end{bmatrix} \tag{15}$$

Note that

$$\begin{bmatrix} Q_{tot} \end{bmatrix}_{22} = \frac{\sigma_0^2 \sigma_z^2 + \sigma_z^2 \sigma_u^2 + \sigma_0^2 \sigma_u^2}{\sigma_0^2 + \sigma_z^2} = \frac{\sigma_0^2 \sigma_z^2}{\sigma_0^2 + \sigma_z^2} + \sigma_u^2 = \sigma_{1|0,z}^2 \tag{16}$$

Note that incrementing the information matrix J with a new incremental entity of information such as a measurement is trivial as illustrated by this example. The only processing problem is that J has to be eventually inverted to extract the actual variances of the state variables. The other issue is of course that of the underlying assumption of the problem being jointly Gaussian. Note that if the measurement provides no information then $\sigma_z^2 \to \infty$ and $\sigma_{1|0,z}^2 = \sigma_0^2 + \sigma_u^2$ as in the previous example.

Next suppose that independent measurements of the anchor were done from both x_0 and x_1 which is illustrated in Figure 2-9.
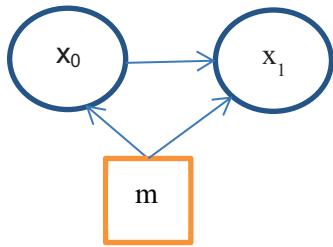
Figure 2-9 *Two independent measurements of the position states x_0 and x_1*

The information matrix for the two independent measurements is given as

$$J_z = \begin{bmatrix} \sigma_z^{-2} & 0 \\ 0 & \sigma_z^{-2} \end{bmatrix} \tag{17}$$

Note that we are implying here that the noise or uncertainty in the measurements is independent and hence the contribution to J_z is a diagonal matrix. The state covariance is then determined as

$$Q_{tot} = J_{tot}^{-1} = \left(Q^{-1} + \begin{bmatrix} \sigma_z^{-2} & 0 \\ 0 & \sigma_z^{-2} \end{bmatrix} \right)^{-1} \tag{18}$$

Next consider the problem depicted in Figure 4-9 but now assume that the anchor point is from a feature point of initially unknown location. Now we have a state vector of $q = \begin{bmatrix} x_0 & x_1 & m \end{bmatrix}^T$. The covariance matrix of the updates is given as

$$Q = \begin{bmatrix} \sigma_0^2 & \sigma_0^2 & 0 \\ \sigma_0^2 & \sigma_0^2 + \sigma_u^2 & 0 \\ 0 & 0 & \sigma_m^2 \end{bmatrix} \tag{19}$$

The information added by the two independent measurements between m and x_0 and between m and x_1 are captured in the pair of information matrices as

$$J_0 = \begin{bmatrix} \dfrac{1}{\sigma_z^2} & 0 & -\dfrac{1}{\sigma_z^2} \\ 0 & 0 & 0 \\ -\dfrac{1}{\sigma_z^2} & 0 & \dfrac{1}{\sigma_z^2} \end{bmatrix} \tag{20}$$

and

$$J_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{1}{\sigma_z^2} & -\dfrac{1}{\sigma_z^2} \\ 0 & -\dfrac{1}{\sigma_z^2} & \dfrac{1}{\sigma_z^2} \end{bmatrix} \tag{21}$$

Note that while the two measurements are independent they individually couple the state variables of m and x_0 and m and x_1. Hence the overall covariance matrix is given as

$$Q = \left(\begin{bmatrix} \sigma_0^2 & \sigma_0^2 & 0 \\ \sigma_0^2 & \sigma_0^2 + \sigma_u^2 & 0 \\ 0 & 0 & \sigma_m^2 \end{bmatrix}^{-1} + \begin{bmatrix} \dfrac{1}{\sigma_z^2} & 0 & -\dfrac{1}{\sigma_z^2} \\ 0 & \dfrac{1}{\sigma_z^2} & -\dfrac{1}{\sigma_z^2} \\ -\dfrac{1}{\sigma_z^2} & -\dfrac{1}{\sigma_z^2} & \dfrac{2}{\sigma_z^2} \end{bmatrix} \right)^{-1} \tag{22}$$

An interesting observation in this example is that it is the multiple connectivity of a fp that makes it carry useful information. A fp measurement that is only connected to one state variable does not convey information. We can see this in the present example from the observation that

$$
\begin{vmatrix} \dfrac{1}{\sigma_z^2} & -\dfrac{1}{\sigma_z^2} \\[3mm] -\dfrac{1}{\sigma_z^2} & \dfrac{1}{\sigma_z^2} \end{vmatrix} = 0
$$

Also it should be pointed out that the component of the information matrix for the measurement has the form of

$$
[J_0]_{ij} = E\left[\frac{\partial^2}{\partial q_i \partial q_j} \ln p(z\,|\,q) \right] \tag{23}
$$

where $p(z\,|\,q)$ is the joint pdf of the measurements. Note that for the case where the measurements are jointly Gaussian with a covariance that is independent of q then

$$
[J_0]_{ij} = 2 \frac{\partial E[z]^T}{\partial q_i} Q_z \frac{\partial E[z]}{\partial q_j} \tag{24}
$$

Reconsider the one step problem shown in Figure 4-6 with the assumption that initially nothing is known of the position of x_0 and x_1. Now information is given as $x_0 \sim N(0,\sigma_0^2)$ and $x_1 - x_0 \sim N(0,\sigma_u^2)$. We then have

$$
Q_s = \left(\begin{bmatrix} \infty & 0 \\ 0 & \infty \end{bmatrix}^{-1} + \begin{bmatrix} \sigma_0^{-2} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \sigma_u^{-2} & -\sigma_u^{-2} \\ -\sigma_u^{-2} & \sigma_u^{-2} \end{bmatrix} \right)^{-1}
$$
$$
= \begin{bmatrix} \sigma_0^2 & \sigma_0^2 \\ \sigma_0^2 & \sigma_0^2 + \sigma_u^2 \end{bmatrix} \tag{25}
$$

In the special case where the following conditions are met:

1. initial distribution of the state variables is jointly normal

2. update of the state vector is a linear function of the state at the previous iteration

3. measurements relate linearly to the state variables

then the state variables conditioned on the measurements remain jointly normal. In this case the MMSE estimate of the state variables given as the conditional mean as (Kay pg.382)

$$
q_{mmse} = E[q] + Q_{qz} Q_{zz}^{-1} z \tag{26}
$$

is the efficient estimator. The covariance of this estimator is denoted as $Q_{q|z}$ and is given as

$$Q_{q|z} = Q_{qq} - Q_{qz}Q_{zz}^{-1}Q_{qz}^{\ T} \tag{27}$$

where $Q_{q|z}$ is the inverse of the Fisher Information matrix. Hence the information matrix provides a direct quantification of the variance of the state variables without having to calculate the estimator first. Hence the information matrix provides an important means of determining the classes of SGL SLAM problems that are solvable.

The information matrix is the kernel of any Bayesian filter approach and especially SLAM which incorporates many disparate observations. Recognizing the information associated with each observation such that it contributes constructively to the estimation of the eventual overall set of state variables is the key to a SLAM implementation. Details of SLAM based on various Bayesian filters will be given in part B.

2.3 Downlink LTE signalling

In this section we intend to expound on the details of the downlink of an LTE system. That is the link between eNodeB to UE. For convenience we also denote the eNodeB as the access node or AN. LTE also known as *evolved universal terrestrial radio access* (E-UTRA), Super 3G (S3G) has differentiated itself from WCDMA by using Orthogonal Frequency Division Multiplexing (OFDM) as a way to transfer data for its downlink scheme and Single Carrier FDMA (Frequency Division Multiple Access) for its uplink scheme.

LTE is designed to transfer packets and should satisfy the following specifications.

- 100 Mbit/s for downlink and 50Mbit/s for uplink
- Optimized operation for mobility of 0 to 15 Km/h and high performance between 15Km/h to 120 Km/h and connected at speeds above 120 Km/h till 350 Km/h.
- Bandwidths of 1.5 to 20 MHz depending
- Support of both FDD (Frequency Division Duplexing) and TDD (Time division Duplexing)
- User latency of 5ms and control latency of 100 ms going from idle to active
- Support for 20 users at 5MHz bandwidth and 400 users at higher bandwidths.

The difference between the LTE and prior standards is the reduction of latency and new modulation scheme that has improved the performance. LTE has reduced the interfaces between the core network and the base station which is also called eNodeB. In LTE eNodeB

does the both jobs of mobility management and radio resource management which reduce the latency between these tasks. Figure 2-6 shows the structure of LTE architecture.

LTE
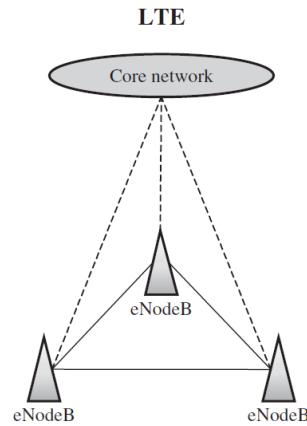
Figure 2-6 LTE architecture

LTE is divided into user plane protocols and control protocols which all reside between the eNodeB and the UE (user end) or as it called the mobile terminal (also User Equipment). These protocols are shown in Figure 2-7.
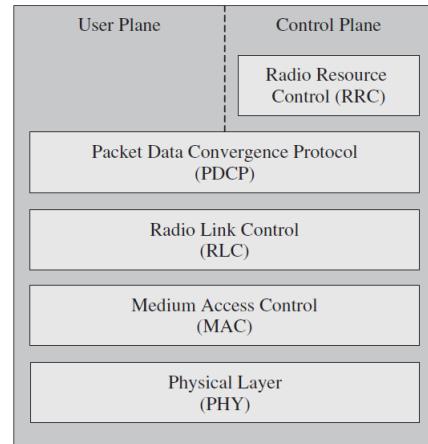
Figure 2-7 User plane and Control plane protocols

The protocol layer we are interested in is the Physical layer which is all the modulation; coding and antenna schemes are located. LTE uses OFDM with cyclic prefix. OFDM is a type of multi-carrier transmission where narrowband channels overlap each other but remain orthogonal due to their unique orthogonal basis function. Figure 3-3 shows an OFDM carrier division. Figure 2-8 shows LTE block diagram.
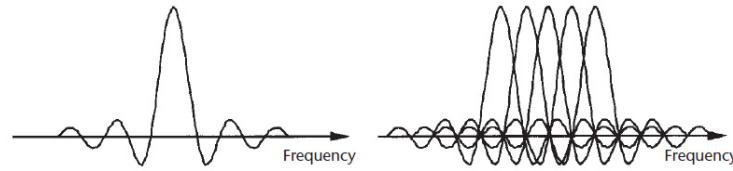
Figure 2-8 OFDM carrier spectrum
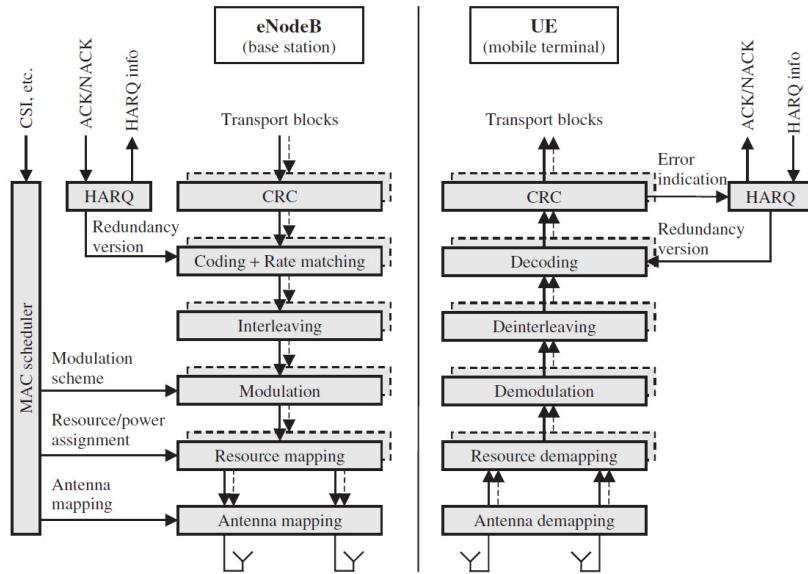
Figure 2-9 LTE Physical layer block diagram

The two important blocks that we are interested in are the resource mapping and the antenna port blocks. Resource mapping has the task to assign symbols to the carriers based on the quality of the particular channel. LTE transmission is based on radio frames. LTE has a 10 ms radio frame which is divided into 10 sub-frame and each 1 ms sub-frame is divide into 2 slots of 0.5 ms. Slot is the smallest unit of LTE radio structure with 0.5 ms time duration each slot contains 6 or 7 symbols for normal and extended cyclic prefix consequently. The length of cyclic prefix is related to the channel length and the symbol number. A larger CP is intended in rural areas and the shorter CP in urban areas where the channel length is smaller and higher throughput without a loss of signal orthogonality is required.

OFDM symbols have either normal or extended cyclic prefix. In case of a normal CP, the CP length is equal to 5.2 us for the first symbol and 4.7 us for the remaining symbols. A CP length of 17.7 us is used for the extended CP which as mentioned before is used in areas where there is a long delay spread. OFDM symbol duration without a guard interval has a duration of 66.7 us. Figure 2-10. Shows LTE radio frame and the mentioned timings.
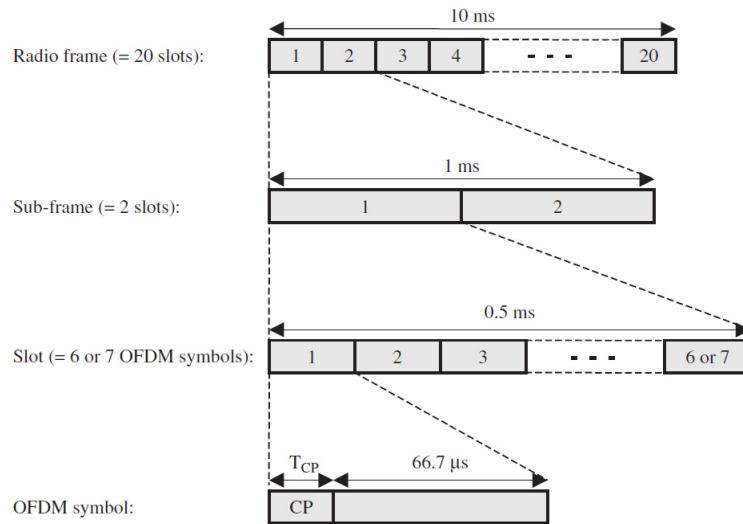
Figure 2-10. LTE radio frame

LTE supports channel bandwidth from 1.4 to 20 MHz and subsequently subcarrier sizes from 128 to 2048 carriers. Sampling time of it has been carefully placed at multiple of 3.84 MHz at 30.72 MHz for back compatibility with legacy standard like UMTS and EDGE.

The subcarrier frequency space is divided into smaller units named as resource block. Each resource block consists of 12 carriers and 2 slots which can virtually be shown in time and frequency domain. A resource block consists of 84 resource elements for normal cyclic prefix and 72 resource elements for extended cyclic prefix. A resource element is a symbol transmitted on a single carrier on an OFDM symbol. Carriers in a resource block are spaced 15 KHz apart, therefore an RB spans the bandwidth of 180 KHz.

In order to obtain channel information for coherent detection and synchronization some carrier-symbol cells of the resource block is dedicated to reference signals. These reference

signals are either Pilots, which are known symbols transmitted at certain RE's or dedicated channels for synchronization such as PSS (primary sync. series) and SSS (secondary sync. series) channels. Pilots are inserted in the RB depending on what antenna port is being used.   In a simple single antenna scheme pilots is placed in
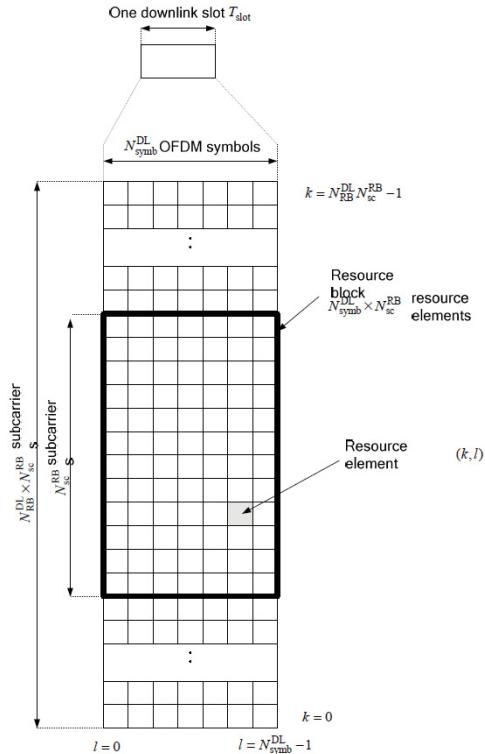
Figure 2-11 Resource block (RB) and Resource Element (RE)

first and third last OFDM symbol. Pilots are placed 6 carriers apart in frequency domain and 3 symbols apart in time domain. The simplest channel estimator scheme needs to interpolate in time and frequency domain to obtain the channel response for the entire resource block. Obviously more sophisticated methods are used when mobility is involved.
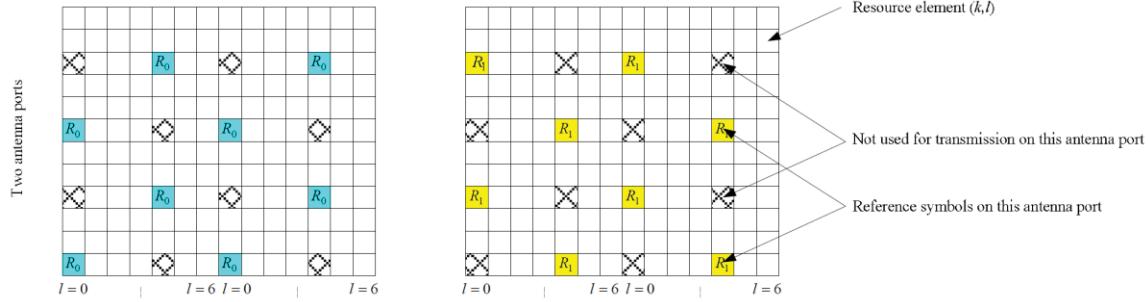
Figure 2-12 Resource element and pilot positioning in CP for two antenna ports

In a multiple antenna scheme pilots are positioned such that they are orthogonal to other antenna ports. A channel estimation algorithm requires calculating the channel information for each of the antennas separately. In a multiple antenna scheme, the resource elements occupied for the pilot in one antenna port is empty in another antenna port RE to avoid interference with each other. Figure 2-12 shows the pilot positioning for a normal CP scheme in two antenna ports. In the channel estimation section we will show how the pilots are used for estimating the channel information. Before we proceed into detail physical layer algorithms we need to investigate the procedures in LTE to know how UE and base station communicate.

Important physical layer procedures

The most important procedure that UE uses to initiate a communication with the eNodeB are:

- Cell search
- Random Access procedures

CELL SEARCH

Cell search is the method that UE uses to find a suitable base station to communicate with. UE identifies the cell which is associated with the base station via detecting the primary and secondary synchronization signals that are transmitted from the base station on regular basis. When discussing the synchronization procedures we will examine the cell search procedure in more detail.

What important for us is that cell search procedure is also utilized to search neighbouring cells for handover procedures. Resource have to be allocated at specific time to minimize interference with cells own resources.

RANDOM ACCESS PROCEDURES

UE requests resources from base station using the RA channels. This would be important as if utilized extra information would be available for the UE to estimate its position. In order to initiate such requests the user terminal needs to send known preambles which are the Zadoff-Chu sequences. These sequences have the characteristic that they build an orthogonal basis each and when transmitted together are separable. As long as they are not repeated by different users there would be no collision. When base station detects the preamble or in other words the RA procedure request, the terminal can send further information and initiate the channel. Random access channels can be a part of standalone locationing algorithm as it utilizes the eNodeB and UE alone.

In order to conduct the experiments as realistic to the LTE physical characteristics we briefly outline the LTE characteristics.

SUPPORTED BW

LTE supports the following bandwidths. Each bandwidth then would have specific RB's that it can support which in turn dictates the number of maximum pilots available to each UE. Table 2-1. Shows the available bandwidth.

Channel bandwidth (MHz)	1.4	1.6	3	3.2	5	10	15	20
Number of resource blocks with FDD	6	N/A	15	N/A	25	50	75	100
Number of resource blocks with TDD	6	7	15	16	25	50	75	100

Table 2-1. Supported Bandwidths

LTE DUPLEX

LTE transmits in FDD or TDD full and half duplex method. E-UTRA TDD band 40 is the band that occupies the ISM band and could be utilized for tests. This band occupies 2300 to 2400 MHz.

LTE SPECTRAL MASK

FCC guidelines for the LTE transmission masks are shown in Figure 8. It is suggested that tighter requirements be placed for locationing purposes to reduce interference from neighbouring cells. Possibly this should go with better pulse shaped OFDM transmitter and receivers for locationing purposes. This is feasible if non-commercial radio is intended.
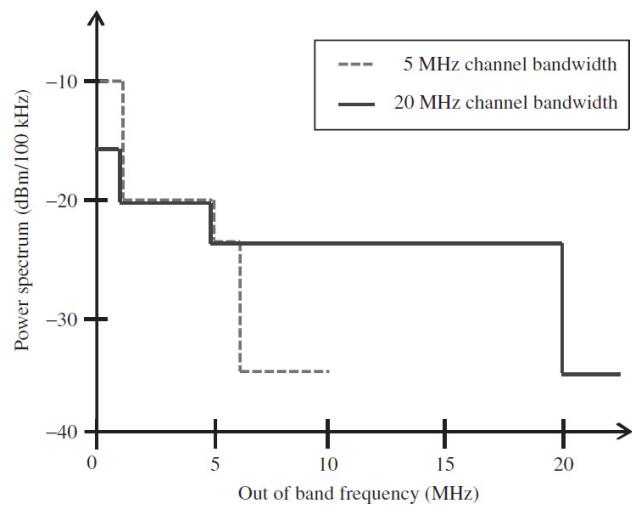
Figure 2-13 LTE TX spectral mask

OFDM have succeeded as a high data rate transmission scheme on the fact that it combines high data rates with low symbol period time. Unlike a single carrier transmission where data rate is inversely proportional to the symbol duration T_s, therefore susceptible to channel delay spread T_d, OFDM by transferring serial data symbols in parallel increases he symbol time in expense of slight frequency bandwidth increase.

In OFDM stream of data is first converted from serial to a parallel stream and then transmitted onto each carrier. The transformation of serial stream into N parallel data streams increases the symbol time, which is the time it takes for one symbol to transmit, by a factor of N. Figure 9. shows a typical OFDM transmission scheme.
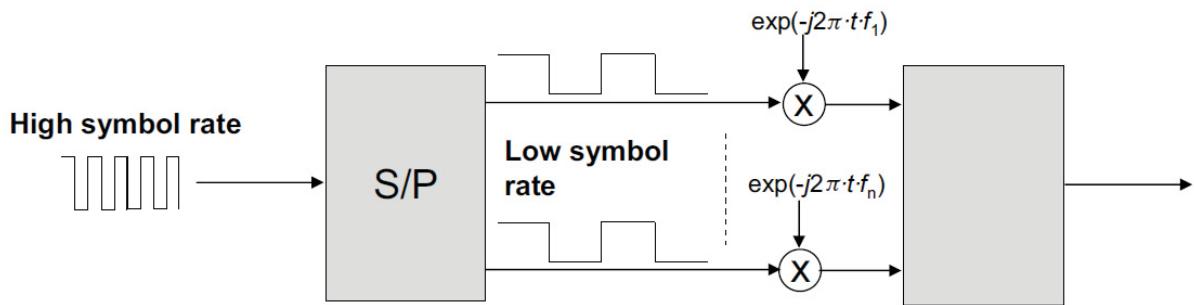
Figure 2-14 OFDM transmission

As seen in Figure 2-14 each symbol is transmitted via one carrier. In an OFDM system total number of carriers is always more than the transmitted symbols. Some of these extra carriers are used as guard bands and some used to transmit reference symbols. Figure 3-10 shows an OFDM transmitter and receiver.
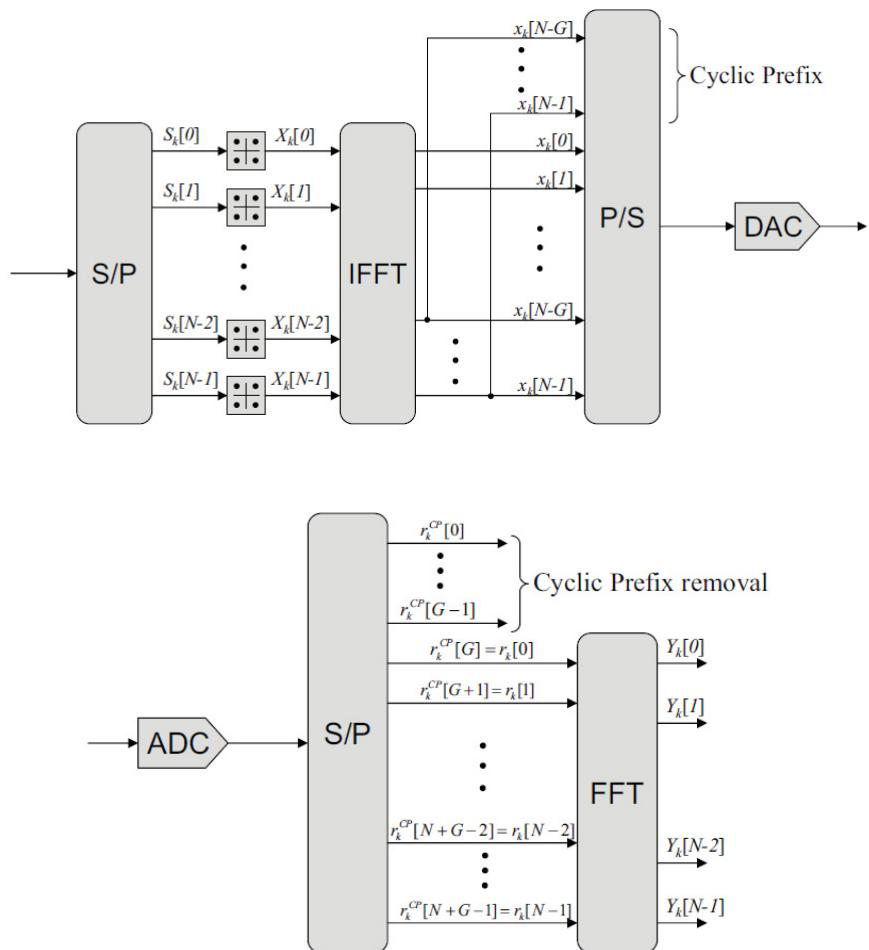
Figure 2-16. OFDM transmitter and receiver

One concept in modem design is the equalization and match filtering. OFDM modulation scheme has solved the match filter concept by using two orthogonal and matched basis function which is sine and cosine transformation. An iFFT and FFT is used as transmitter and receiver matched filters. In such scheme equalization of the signal is done in frequency domain, therefore prior to match filtering preserving the symbol energy in one FFT block is very important.

OFDM solves the channel effect and the ISI by introducing Cyclic Prefix and by simply copying the last L symbols to the beginning of the block and then transmitting through the channel. By doing so any channel with length less than L will have its effect preserved in the CP. In this way an FFT block becomes cyclo-stationary and it would be unimportant from which position the FFT block is taken as long as it contains the whole CP as a part of the block. By doing this basically the effect of the channel is reversed and OFDM blocks are capable of equalizing the channel very effectively with minimal resources provided that they are given the channel information in frequency domain. Figure 3-11. shows the position of CP in an FFT block. In this figure T_{cp} is the duration of the CP which depends on the channel it is going to be used. T_u is the actual symbol duration.
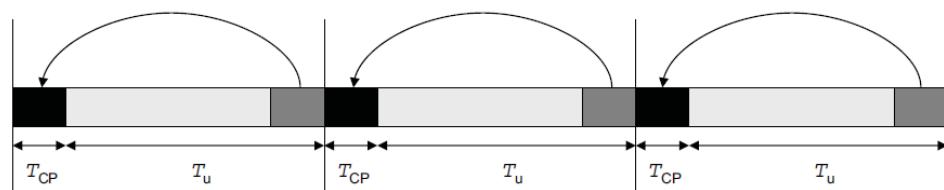
Figure 2-17: Cyclic Prefix in FFT block

At the receiver the beginning of the packet is found and from that position and FFT block is separated and an ISI free block is passed to the FFT which in turn transfers back the signal from what is known time domain to frequency domain. We will dig into the math of such operation next.

Down Link Model

Now that we have familiarized ourselves with OFDM, LTE basic facts and procedures we will discuss the LTE downlink specifically.

- LTE resources

- LTE Synchronization

- LTE Channel estimation

We also discuss LTE preambles (Zadoff-Chu sequence) and their unique characteristics.

SYNCHRONIZATION

No resource can be extracted including pilots if proper synchronization is not performed between eNodeB and UE. The following synchronization has to be performed:

- Carrier frequency synchronization
- Symbol and frame timing synchronization
- Sampling clock synchronization

Without the above three synchronization no pilots can be extracted and no channel information culd be calculated, therefore prior to channel estimation we are interested in how to make a good synchronization.

CELL SEARCH

UE (user equipment) needs to perform a cell search procedure to access an LTE cell. This involves detecting the Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS) which is broadcasted by the eNodeB at specific radio frame locations for this purpose.
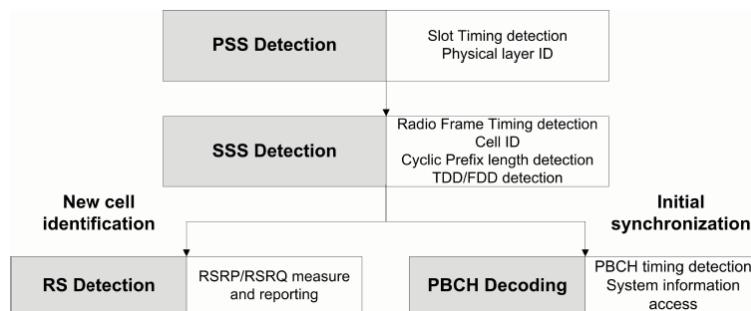
Figure 2-18: Cell search procedures

As seen from Figure 2-18, in order to examine the pilots for neighbouring cell and associated cells, SSS and PSS has to be performed in order. In a radio frame the PSS and SSS signal position is different for FDD or TDD transmission. Figure 2-19 shows the position of these signals in the

radio frame for both types of frames. Prior to SSS detection the CP length is unknown so the synchronizer blindly should search for the SSS in the given locations. Totally 4 different positions has to be searched if both TDD and FDD is supported in an algorithm design. PSS signal always proceeds SSS either immediately (FDD) or after 3 symbols (TDD). The job of a locationing algorithm besides extracting information from the channel information is to provide better synchronization algorithms for neighbouring cells if it intends to use multiple cell information in its positioning algorithms.

We mentioned the position of the PSS and SSS signals in time domain which their position in the radio frames. Now we are interested in their position in frequency domain. As no bandwidth information is known prior to synchronization the PSS and SS are positioned in centre of the frequency band occupy 6 resource blocks (RB's) which is the minimum number to allocate. The PSS and SSS have 62 symbols which are all mapped to 62 central carriers in the 6 mentioned RB's. As mentioned before each RB has 12 subcarriers which mean from 72 subcarriers available only 62 carriers are used and the rest is not used. This means 5 RE at each end of the resource block is not used. Figure 2-20 shows the position of the RB's and the empty RE's at the lower end RB.

(a)   (b)
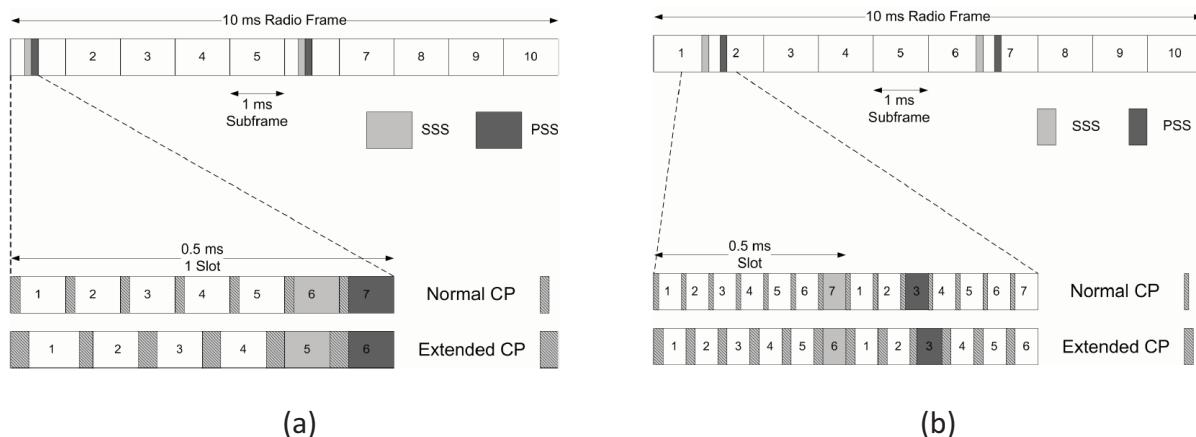
Figure 2-19: (a) FDD frame, (b) TDD frame
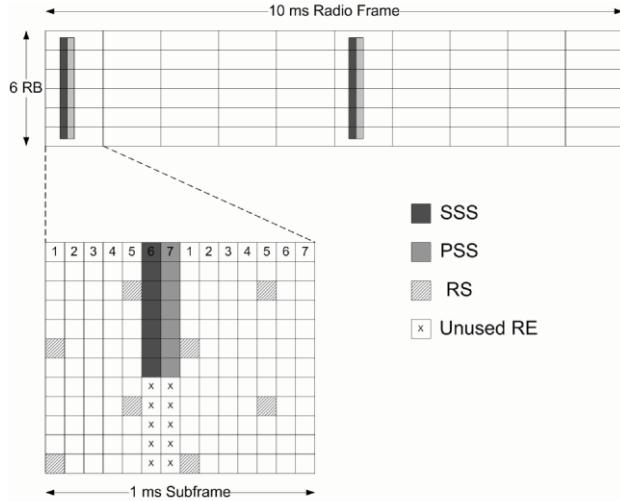
Figure 2-20: PSS and RSS position in RB's

DETECTION OF PSS AND SSS

PSS uses Zadoff-Chu (ZC) sequence. ZC sequences are Constant Amplitude Zero Auto correlation sequences. Unlike golden codes these codes are non-binary and suitable for IQ transmission. Equation 1 shows the odd length ZC sequence:

$$a_q(n) = e^{\left\{-j2\pi q\left(\frac{n(n+1)}{2}\right)+l.n\middle/N_{ZC}\right\}} \qquad (1)$$

Where :

$q\epsilon\{1, \dots, N_{ZC} - 1\}$ is the root index.

$n = \{0,1, \dots, N_{ZC} - 1\}$

$l\epsilon$N where in LTE l = 0 is used.

$$R_{kk(\tau)} = \sum_{n=0}^{N_{zc}-1} a_k(n)a_k^*(n + \tau) = \delta(\tau) \qquad (2)$$

Where R(.) is the autocorrelation function of a_k at position τ

The properties of these sequences are:

- ZC sequences have constant amplitude in time and frequency domain.

- ZC codes have ideal cyclic autocorrelation which enables the user to detect the codes even when they are delayed. If the code is designed to operate at delay N. The autocorrelation of a delayed sequence will generate a peak at the delayed time as long as the delay is shorter than the designed maximum delay of N samples.
- Multiple orthogonal sequences can be generated from one sequence. The cross correlation of two sequences have a constant amplitude of $1/\sqrt{N_{ZC}}$ if and only if the difference of sequence indices are relatively prime to N_{ZC}. This means if N_{ZC} is chosen as a prime number $N_{ZC} - 1$ orthogonal sequences may be generated by just changing the indices of the sequence.
- A DFT of the ZC sequence is a weighted cyclic shift ZC. This means ZC can be generated in frequency domain directly. Later some papers will be shown that use this property to generate ZC sequences efficiently.

PSS SIGNAL:

As mentioned above the ZC codes can be generated in frequency domain directly. The frequency domain codes have the exact same characteristic as time domain ZC codes. A PSS signal uses this feature and transmits a 63 length ZC code on 63 carriers with middle code punctured to avoid dc subcarrier. LTE uses 3 PSS codes for 3 possible physical layer sections in a cell which means one PSS code for each group in the cell.

In equation (1) q is the root of the ZC sequence. PSS codes are defined if choosing $l = 0$ and q = 29, 34, 25. The chosen PSS signals have low sensitivity on frequency offset and can still be detected if there is a 7.5 KHz frequency drift. This is due to their almost flat frequency response. A correlator will decide which code has been transmitted. The received signal is correlated against the existing codes and the position of the peak shows the offset of the signal.

SSS SIGNAL:

SSS is signal is an M-sequence. It is constructed by interleaving two 31 length BPSK M-sequence codes known as SSC1 And SSC2. SSC1 and SSC2 are cyclic shifted versions of a 31 length M-sequence. The shift of each code is mentioned in 6.11.2.1-1 in technical specification 36.211 of 3GPP group. SSC2 is scrambled by a sequence depending on the SSC1 shift and SSC1 is scrambled based on the PSS signal index.

SSS detection is done after PSS detection and therefore the system has the channel information at this point. The system can either use a coherent detector or a non-coherent one which is based on the code correlation.

In coherent detection the performance depends on the presence of the interferer and the quality of the channel estimation, this is due to the fact that the interfere uses the same PSS signal therefore the SC1 and SC2 signals would be identical. In a non coherent detection the performance depends on the coherent bandwidth of the channel which should be greater than the 6 resource blocks being used by the SSS signal. This ensures that the SS signal would not be changing during the transmission.

Channel estimation

A locationing algorithm needs channel estimation. A coherent detector is used that can utilize the phase and amplitude of the received signal. A common way to estimate the channel information is to use a known signal such as a pilot. These known signals are called Reference Signals (RS's). One can realize an OFDMA signal as a three dimensional frame of time, frequency and space signal. RS signals are placed in the LTE frame at known time and frequency positions on an optional or chosen space which is the antenna port. The position in frequency is called subcarrier and the position in time is known as slots and the smallest element in this two dimensional domain is called the Resource Element (RE).

LTE downlink uses 5 different reference signals (RS) based on the LTE release number.

- Common reference signal which are available to all UEs (*Pilots*)
- Demodulation reference signal specific to the UE
- Multimedia broadcast Single Frequency Network MBSFN specific reference signal
- Positioning reference signal (Release 9 onwards)
- Channel state Information reference signal (Release 10 LTE advanced)

Reference signals are transmitted from eNodeB. LTE introduces antenna ports as ports to transmit RS which can either be physical antenna or virtual ports. From UE point of view any measurement obtained from these antenna ports defines the characteristic of that port.

What we are going to first target for the channel estimation is the common reference signals or the cell specific signals. These RS signals are available to all UE's in the cell and it is a standard RS since release 8, these are the primary RS for channel estimation purposes.

In Figure 15 an R_0 indicates a RS for antenna port zero. The transmitted RS signal is a QPSK signal defined in equation (3).

$$r_{l,n_s} = \frac{1}{\sqrt{2}}[1 - 2c(2m)] + j\frac{1}{\sqrt{2}}[1 - 2c(2m+1)] \qquad (3)$$

In equation (3), m is the reference signal index, n_s is the slot number and l is the symbol number. C(m) is taken from a length 31 gold sequence which is initialized based on the RS type. The initialization is done every OFDM symbol and the value depends on the cell identity, N_{ID}^{cell}. Each cell uses a cell specific frequency shift equal to $N_{ID}^{cell} \bmod 6$ which avoids frequency collisions between the RS signals from neighboring 6 cells. This is an important feature when using the channel estimation from neighboring cells to calculate the location.
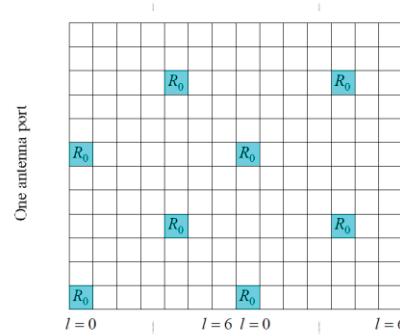
Figure 2-21: Cell specific RS for normal CP length

We would leave the rest of the RS signals for now till later phases of the project but before leaving this section we need to mention the next important RS which is the UE specific RS. UE-specific RS are used only between eNodeB and particular UE and are transmitted as of any other PDSCH (Physical Downlink Shared Channel) signal. The difference is that these RS signals are mapped to specific RB's that target the specific UE and therefore contain UE specific RS signals. This can be beam-forming signals or extra RS signals. We will go into details of these specific RS signals later.

We will consider the channel $h(\tau, t)$ to be the channel time varying complex signal at delay τ and time instant of t. In a discrete form this channel is modelled as $h[l, kT]$ with l as the delay

at time sampling instant kT. We also consider the channel to be approximated by an FIR vector of:

$$h[k] = h[0, k], \ldots, h[L - 1, k] \qquad (4)$$

where L is the channel delay.

Practical channel estimation for LTE is done in frequency domain. If the received signal is:

$$y = h * x + n \qquad (5)$$

The FFT of the received signal is:

$$Y = HX + N \qquad (6)$$

Where X is the Nx1 FFT of the transmitted symbols and H is the LxN cyclic FFT of the channel information and N is a zero mean complex cyclic white noise with covariance of C. In order to obtain the channel matrix H it is sufficient to correlate the X with the known transmitted symbols and obtain H:

$$Y = H(X.X^H) + NX^H = H + z = \widehat{H} \qquad (7)$$

If X is the reference symbols and has zero mean and unit energy then z is still a circular zero mean noise and the result would be the NxL estimated channel where L is the channel length. As shown in Figure 15, the channel estimation would result in estimating the channel for locations in frequency-time lattice that the RS signal is present. In order to estimate the channel for the rest of the subcarriers approximate estimation of the channel at the frequency and time lattice is required. If the frequency and time is considered statistically independent then we can divide the estimation in 2 domains as a single domain linear estimation which in its simplest form will be interpolation.

More sophisticated approaches to obtain the two dimensional estimates would be LS least square and MMSE approach. Any approach in time domain has been proven to be impractical although from estimation point of view it would have resulted in more accurate unbiased estimations. Chapter 8.4 in [1] compares different channel estimation methods but the most practical channel estimation for an immobile user is the interpolation method described above.

Simulation Environment

The system requires TOA and channel estimation from multiple eNodeB's at the UE position. As mentioned in previous sections the synchronization methods provided in LTE system would provide the TOA of signals from different cells to the UE and the channel estimation methods described in previous section would provide the channel information needed. In order to set up such simulation environment a single antenna downlink scheme as shown in Figure 16 is provided:
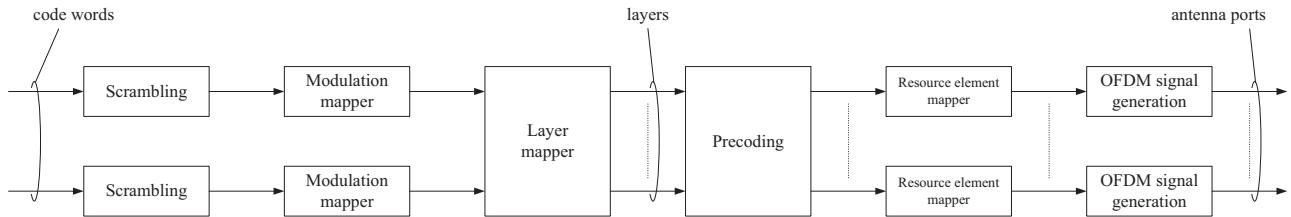
Figure 2-22. Downlink Physical Chain per eNodeB

The OFDM signal from different eNodeB goes through a Lagrange interpolator to change phase according to their distance to the UE and then multiplexed with other eNodeB's and passed through the channel. Figure 2-23 shows an alternative that the signal is individually passed through the channel and delay is performed on the signal after the channel.
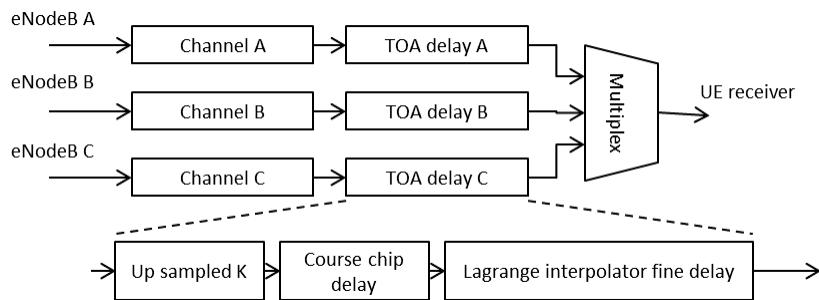
Figure 2-23. Downlink channel

On the receiver the PSS and the SSS signal is extracted and the delay associate with the intended eNodeB is considered. The interpolated signal for each intended eNodeB is passed to their channel estimator and the channel is extracted for each eNodeB-UE path.

The LTE observables which are relevant are essentially the known synchronization signal segments that are part of the resource blocks. While it is a reasonable assumption that the LTE AN signals are frequency synchronized (which can be accurately obtained from GPS signals amongst other potential sources), it is not a reasonable assumption in many cases to assume that the AN's are accurately time synchronized. This depends as the network may be specifically calibrated for SGL purposes. Tight timing synchronization amongst the AN's is not necessary for optimum data communication functions nor is it necessary for facilitating network based location, hence there is little impetus for the service provided to accurately time synchronize the network. Consequently, the SGL processing developed will consider the two modes when the LTE network is tightly time synchronized and when it is not.

Chapter 3   CV observables and processing

3.1 Overview

The SGL observables created from LTE range measurements are useful as the wireless signals are generally ubiquitous throughout the environment that the UE is immersed in. The LTE AN's are generally frequency synchronized but may not be time synchronized and with possible uncertainty in terms of the AN transmitter location. Furthermore the LTE signals are subject to a significant amount of multipath which for isolated SGL estimates renders them virtually useless. However the SLAM algorithm, developed in the previous chapter estimates the trajectory of the UE in light of all of these uncertainties partially ameliorating the detrimental uncertainty due to multipath. CV observables, complement the LTE observables in that they provide high accuracy over shorter segments of the trajectory. Hence, relative updates in the UE position can be achieved based on CV observables. They also have the very important attribute of being independent of any LTE signalling providing observables where LTE downlink signals are not available or compromised in some way. Additional notable advantages of incorporating CV is that it is possible to detect if the LTE signals are LOS or NLOS over small trajectory segments. Detection of a small LOS trajectory segment implies that beamforming can be used on the LTE signals which are highly effective in determining the relative position of the UE with respect to the AN. Another advantage of using CV is that these observables are immune to LTE signal jamming or spoofing.   Finally CV observables can be sourced from small uncalibrated cameras which are typically part of the UE handset such that additional hardware to generate the CV observables is negligible.

The basic method of generating CV sensor inputs is to track features on the ground manifold surface updating outputs of the differentials in the pose and the translational motion relative to the ground manifold at the frame rate interval of the camera as illustrated in Figure 3.1. The CV update rate of between 10 to 30 frames per second (fps) is possible while maintaining real time processing which is sufficiently fast to provide highly accurate relative updates of the UE position and orientation. Ego-motion is essentially the process of determining the relative position and orientation of the camera based on what it observes. What is observed is the

surrounding 3D scenery in the world reference that is mapped onto the image plane of the camera. If features of known identity and location (relative to the world reference) can be observed then we have ego-motion in the absolute sense. However, a more common scenario is that the features cannot be uniquely classified or do not have a location that is known absolutely. Hence only relative motion and orientation of the camera is possible. In the latter case, it is generally assumed that the feature points are static relative to the world reference and hence the relative ego-motion is possible. In using these static feature points of unknown absolute location, the ego-motion algorithm actually simultaneously determines its own trajectory relative to the array of feature points such that all the observations are consistent. That is, in the end, the relative positioning of both the trajectory and fp's is then established which is a form of SLAM, (simultaneous location and mapping). If a set of known fp's with known locations in the world reference is observed by the camera then the combined array of fp's and trajectory can then fixed absolutely to the world reference. Hence absolute mapping and location is achieved.

Feature points that are suitable for tracking are determined by the GF2T (Good Features to Track) and then tracking them frame to frame with LKP (Lucas Kanade Pyramid) method. The LKP is used to establish the necessary correspondence between the fp's in different frames such that the motion of the fp's can be determined. The set of fp's used will be assumed to be stationary relative to the world reference frame. We can assume different sets of constraints. For instance, we can assume that the FOV of the camera is confined to the floor surface or the ceiling of a building. We may assume that the fp's are distributed instead in the 3D space but perhaps confined to a set of planes. These planes may be known to intersect at 90 degrees or may be more random. Generally we will find that the camera provides a wealth of observations, however, the ego-motion problem is generally ill-conditioned.   Hence while it is possible to do full 6 DOF ego-motion (full evaluation of **T** and **R**) with a single camera, two cameras or a projector and camera is in the Kinect application is typically required to make the implementation robust.

We will start the ego-motion development with a highly constrained motion in 2D. The 2D problem will involve tracking of stationary fp's coincident with a 2D manifold surface such as the floor surface or a wall. As is illustrated in Figure 3-1, the camera is oriented such that it is looking at a flat 2D manifold surface which could be a ground or floor surface. As the camera is moved across this surface, there will be a flow of features in the field of view which is related to

the motion of the camera itself. This 2D problem is significantly simpler than the 3D case where the camera has to perceive a 3D environment, usually with multiple cameras or correlating multiple views of the same camera. Throughout this chapter we will assume a perfectly calibrated camera that behaves as an ideal pin-hole camera which is reasonable as camera calibration can be assumed as it is a robust procedure.
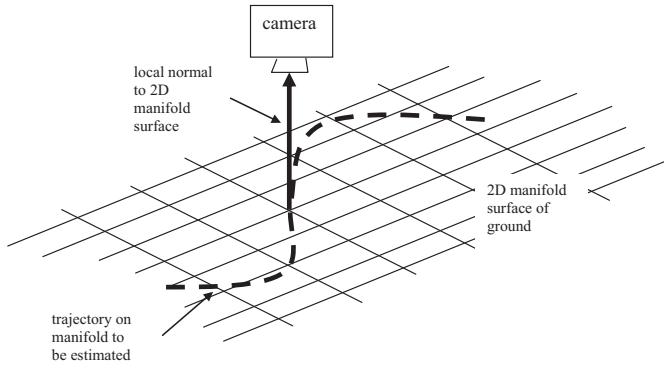
Figure 3-1 *Ego-motion relative to a 2D manifold surface*

3-2 System Definitions

In this section a set of necessary definitions will be given that will facilitate the development of the ego-motion algorithms. Initially only a single camera will be considered. In each image frame made available from this camera as it moves, there is a set of observable fp's. New fp's are typically identified by a corner feature detector. We will denote this as the Good Features two Track (GF2T) algorithm. The features are matched from one frame to the next to determine the correspondence based on the Lucas Kanate Pyramid (LKP) routines.

The set of fp's generated from these algorithms for the tth frame is denoted as $\left\{ f_{k,x}^{t}, f_{k,y}^{t} \right\}$. Here

the index t denotes the frame, k denotes the feature index and $\{x, y\}$ denote the integer indices of the location of the feature point as referenced to the image plane of the camera. The analysis is primarily focussed on a monocular ego-motion single camera video frame sequence consistent with what is in the handheld device. We will start with Ego-motion for a 2D plane as shown in Figure 3-1 followed by ego-motion in a 3D environment. Differential ego-motion will be

considered which is based on determined by how the fp's have moved from the previous camera image to the current image as captured in the fp data $\{f_{k,x}^{t-1}, f_{k,y}^{t-1}\}$ and $\{f_{k,x}^{t}, f_{k,y}^{t}\}$ respectively. From this a differential or incremental update can be determined by relating this to the underlying transformation. As established in the previous chapter on optical flow, finding the correspondence between $\{f_{k,x}^{t}, f_{k,y}^{t}\}$ and $\{f_{k,x}^{t-1}, f_{k,y}^{t-1}\}$ is not trivial. Undetected errors made in the fp correspondences will result in significant distortions of the estimated differential motion update. Parameters for the LKP routine have to be carefully selected to obtain robust fp correspondence assignment. In the following chapter on Bayesian filtering emphasis will be given to the overall uncertainty of the fp migration.

The ego-motion geometry will be based on the world centered and camera centered coordinate systems as described in conjunction with the perspective transformation in a previous chapter with the following definitions:

$\mathbf{O_c}$ - camera origin in world coordinate space

$\mathbf{O_w}$ - world origin in world coordinate space (Usual definition is $\mathbf{O_w} = [0,0,0]^T$)

$f_{w,k}^{t}$ - fp in the world coordinate frame with coordinates of $\{f_{w,k,x}^{t}, f_{w,k,y}^{t}, f_{w,k,z}^{t}\}$

$f_{c,k}^{t}$ - fp in the world coordinate frame with coordinates of $\{f_{c,k,x}^{t}, f_{c,k,y}^{t}, f_{c,k,z}^{t}\}$

$\{\mathbf{X_w}, \mathbf{Y_w}, \mathbf{Z_w}\}$ - world coordinate system directional unit vectors (left hand system)

$\{\mathbf{X_c}, \mathbf{Y_c}, \mathbf{Z_c}\}$ - camera coordinate system directional unit vectors (left hand system)

$\boldsymbol{\pi}$ - image plane, given by the plane $\mathbf{Z_c} = f$

$\{\mathbf{x}, \mathbf{y}\}$ - unit vectors of 2D camera image plane $\boldsymbol{\pi}$

Figure 3- 2 is an illustration of the mapping of a fp between the world and camera coordinate systems. It is important to realize that the rays between $f_{c,k}^{t}$ and $f_{W,k}^{t}$, as illustrated in Figure 3- 2 are essentially lines of ambiguity where the fp actually originates from.
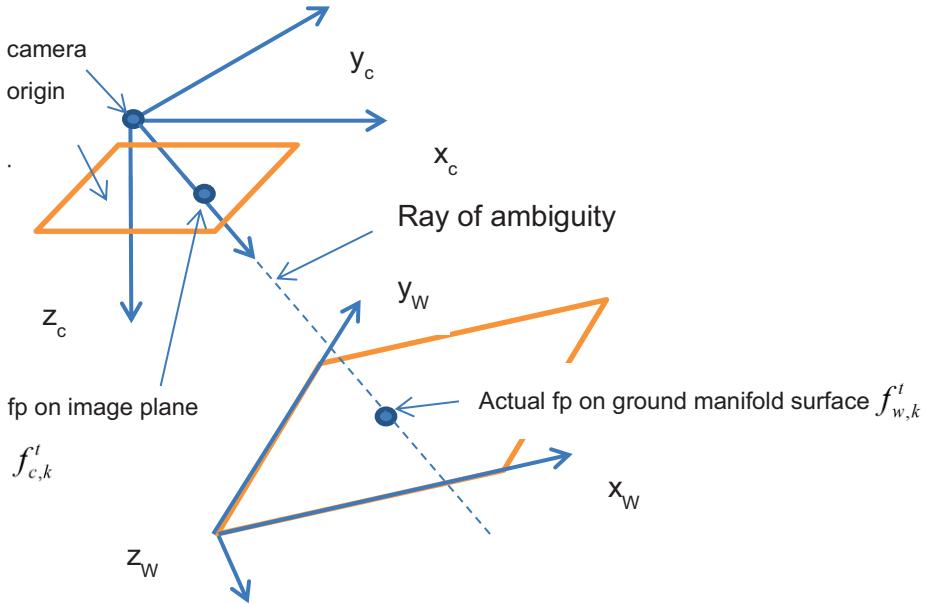
Figure 3-2 *Projection of fp in the world coordinate system onto camera 2D image plane*

The projective transformation defined earlier resulted in the transformation of

$$\mathbf{P_c} = \mathbf{R}(\mathbf{P_w} - \mathbf{T}) \tag{1}$$

relating the position vector in the world coordinate frame to that of the position vector of the camera coordinate frame. Here $\mathbf{T}$ is the translation vector $\mathbf{T} = \mathbf{O_C} - \mathbf{O_W}$ and the rotation matrix is given as the projection coefficients of the unit vectors of $\{\mathbf{X_w}, \mathbf{Y_w}, \mathbf{Z_w}\}$ onto $\{\mathbf{X_c}, \mathbf{Y_c}, \mathbf{Z_c}\}$ as

$$\mathbf{R} = \begin{bmatrix} \mathbf{X_w} \square \mathbf{X_c} & \mathbf{Y_w} \square \mathbf{X_c} & \mathbf{Z_w} \square \mathbf{X_c} \\ \mathbf{X_w} \square \mathbf{Y_c} & \mathbf{Y_w} \square \mathbf{Y_c} & \mathbf{Z_w} \square \mathbf{Y_c} \\ \mathbf{X_w} \square \mathbf{Z_c} & \mathbf{Y_w} \square \mathbf{Z_c} & \mathbf{Z_w} \square \mathbf{Z_c} \end{bmatrix} \tag{2}$$

where $\square$ denotes the dot product between the unit vectors. (1) relates the fp's referenced to the world and camera references as

$$f_{c,k}^{t} = \mathbf{R}\left(f_{W,k}^{t} - \mathbf{T}\right) \tag{3}$$

Expressing $\mathbf{R}$ in terms of row vectors

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3 \end{bmatrix} \tag{4}$$

such that the homogeneous camera coordinates are then

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & -\mathbf{R}_1\mathbf{T} \\ \mathbf{R}_2 & -\mathbf{R}_2\mathbf{T} \\ \mathbf{R}_3 & -\mathbf{R}_3\mathbf{T} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5}$$

From which the non-homogeneous image plane coordinates are written as

$$x = f \frac{\mathbf{R}_1 (\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3 (\mathbf{P}_w - \mathbf{T})}$$
$$y = f \frac{\mathbf{R}_2 (\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3 (\mathbf{P}_w - \mathbf{T})} \tag{6}$$
$$z = f$$

In the application of current interest, a handheld device is hovering over a 2D flat ground surface such that the fp's are residing on the surface the surface of the world reference as $Z_w = 0$ as in figure 1 such that $f^t_{W,k,z} = 0$. We also assume that the fp's are stationary such that for the world coordinates we can drop the "t" index. Then we have

$$\begin{bmatrix} f^t_{c,k,x} \\ f^t_{c,k,y} \\ f^t_{c,k,z} \end{bmatrix} = \mathbf{H} \begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \\ 1 \end{bmatrix} \tag{7}$$

where $\mathbf{H}$ was defined earlier as

$$\mathbf{H} = \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix} = \begin{bmatrix} [\mathbf{R}_1]_1 & [\mathbf{R}_1]_2 \\ [\mathbf{R}_2]_1 & [\mathbf{R}_2]_2 & -\mathbf{RT} \\ [\mathbf{R}_3]_1 & [\mathbf{R}_3]_2 \end{bmatrix} \tag{8}$$

The nonhomogeneous coordinates are then

$$f_{k,x}^t = f_{c,k,x}^t / f_{c,k,z}^t = \frac{H_{11}f_{w,k,x} + H_{12}f_{w,k,y} + H_{13}}{H_{31}f_{w,k,x} + H_{32}f_{w,k,y} + H_{33}}$$

$$f_{k,x}^t = f_{c,k,y}^t / f_{c,k,z}^t = \frac{H_{21}f_{w,k,x} + H_{22}f_{w,k,y} + H_{23}}{H_{31}f_{w,k,x} + H_{32}f_{w,k,y} + H_{33}}$$

(9)

In many cases of practical interest where the fp's as distributed such that $f_{c,k,z}^t$ does not vary significantly then the weak perspective assumption can be used such that $f_{c,k,z}^t$ is taken to be a constant denoted as c_o . Then we have

$$c_o f_{k,x}^t = H_{11}f_{w,k,x}^t + H_{12}f_{w,k,y}^t + H_{13}$$

$$c_o f_{k,x}^t = H_{21}f_{w,k,x}^t + H_{22}f_{w,k,y}^t + H_{23}$$

(10)

Note that we are really implying that $f_{c,k,z}^t$ is constant relative to both time and the fp index k. With this the perspective matrix reduces to that of the affine transformation which is conveniently redefined as

$$\begin{bmatrix} f_{x,k}^t \\ f_{y,k}^t \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} f_{W,x,k} \\ f_{W,y,k} \\ 1 \end{bmatrix}$$

(11)

where $\{a,b,c,d,e,f\}$ are coefficients dependent on the geometry of the camera orientation and position relative to the world reference. This weak perspective transformation or affine transformation is clearly applicable where the camera axis Z_c is collinear with the Z_W axis. That is the camera can translate and rotate in azimuth but have not tilt. The affine transformation is still a reasonable approximation for cases where there is a slight tilt provided that the fp's considered in the world plane of $z_W = 0$ are close to the intercept point of the Z_c axis in the $z_W = 0$ plane.

Finally define $\mathbf{A_t}$ as the transformation corresponding to the tth frame such that

$$\begin{bmatrix} f_{k,x}^t \\ f_{k,y}^t \end{bmatrix} = \mathbf{A_t} \begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \\ 1 \end{bmatrix} \qquad (12)$$

3.3 2D translation ego-motion

Consider the camera oriented horizontally over the $Z_W = 0$ plane as illustrated in figure 3 such that the perspective rotation matrix is $\mathbf{R} = \mathbf{I_3}$. Denote the perspective translation vector as $\mathbf{T}_t$ where the subscript t denotes the frame. For the 2D ego-motion case at a constant height h we have $\mathbf{T}_t = \begin{bmatrix} x_T^t & y_T^t & -h \end{bmatrix}^T$ where $\{x_T^t, y_T^t\}$ is the displacement along the x and y axis of the camera frame relative to the world reference and h denotes the height.

Begin with the simplified 1D case of motion only along the $\mathbf{X_W}$ axis such that $\mathbf{T}_t = \begin{bmatrix} x_T^t & 0 & -h \end{bmatrix}^T$
.   Assume that the sets of feature points of $f_{c,k}^t$ for the current frame and $f_{c,k}^{t-1}$ from the previous frame are available.
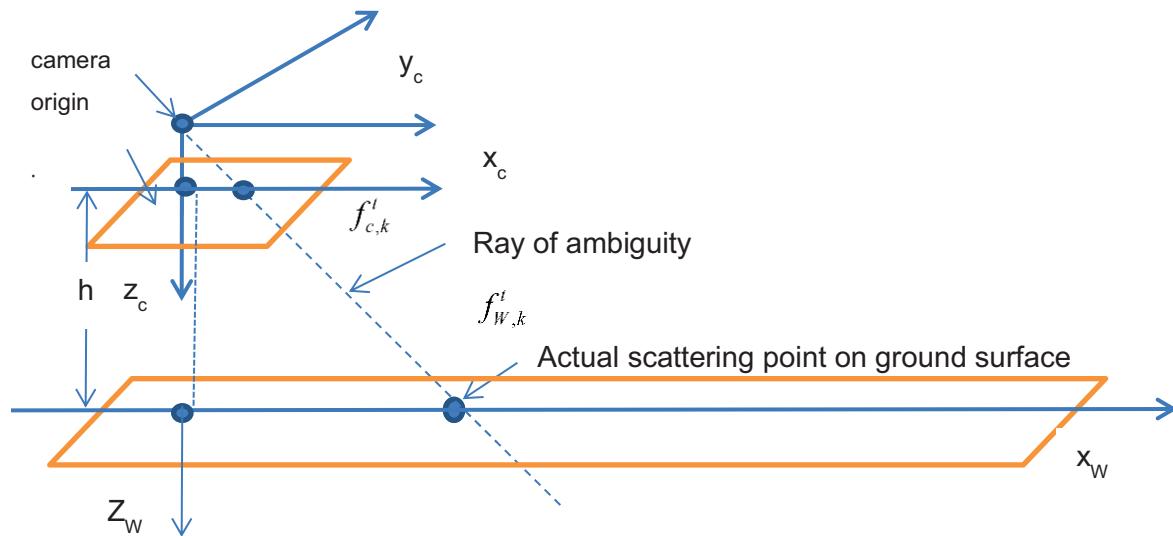
Figure 3 *Simplified 1D translation in the $\mathbf{X_W}$ direction*

Based on this definition the affine transformation is given as

$$\mathbf{A_t} = \frac{f}{h}\begin{bmatrix} 1 & 0 & -x_T^t \\ 0 & 1 & 0 \end{bmatrix} \tag{13}$$

Typically the scaling of f/h is ill defined as the focal length of the camera is typically not well known. Hence we replace this with a scaling factor of s which is assumed to be known from the camera calibration. The overlay of the kth fp in two successive frames is illustrated in Figure 4. Note that we can determine the differential change $\Delta\mathbf{A_t} \Box \mathbf{A_t} - \mathbf{A_{t-1}}$ but we cannot determine the absolute $\mathbf{A_t}$. From (12) we have

$$\begin{bmatrix} f_{k,x}^t \\ f_{k,y}^t \end{bmatrix} - \begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix} = (\mathbf{A_t} - \mathbf{A_{t-1}})\begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \\ 1 \end{bmatrix} = \Delta\mathbf{A_t}\begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \\ 1 \end{bmatrix} \tag{14}$$

Based on the case where we have motion only along the x axis

$$\Delta\mathbf{A_t} = s\begin{bmatrix} 0 & 0 & -\Delta x_T^t \\ 0 & 0 & 0 \end{bmatrix} \tag{15}$$

where $\Delta x_T^t \Box x_T^t - x_T^{t-1}$ and s is a scaling factor. Based on similar triangles we can derive

$$\Delta x_t^T = -\frac{1}{s}\left(f_{k,x}^t - f_{k,x}^{t-1}\right) \tag{16}$$
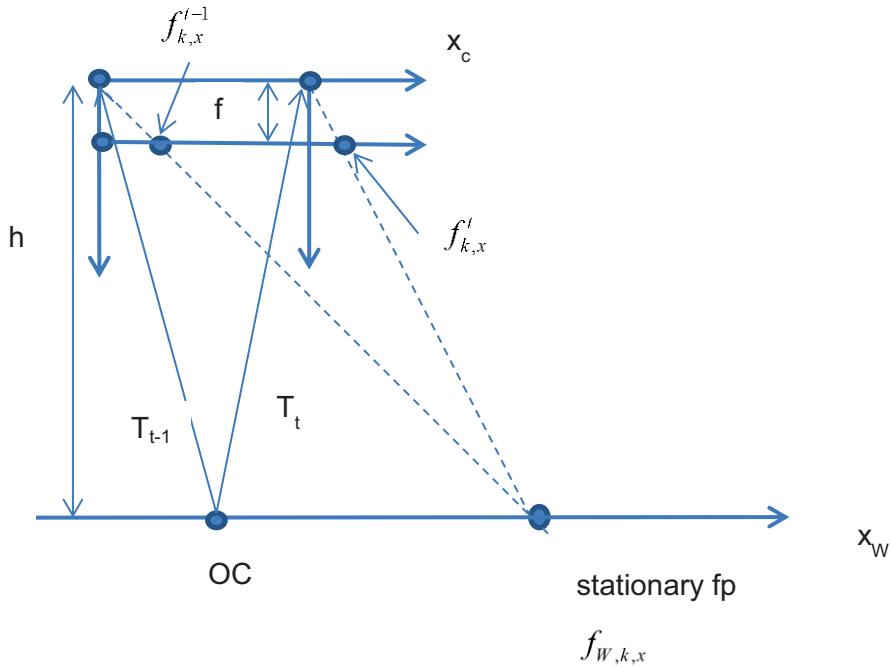
Figure 4 *Similar triangles used to determine the geometry between the camera image plane and ground plane*

Note that we can estimate the relative 1D ego-motion based on a single fp. If the fp is a landmark or anchor point then we can determine the absolute ego-motion from just a single frame with the assumption that the camera is precisely calibrated. As stated earlier, the practicality of CV ego-motion estimation is that the relative camera motion can be determined from arbitrary and previously unknown fp's that are only assumed to be temporally static relative to the world (ground manifold) reference frame. An issue with fp's is that they are subject to random fluctuations from one frame to the next as observed in the previous chapter. Typically the random fluctuation is independent from one fp to the next. Hence the random deviation of Δx_T^t can be reduced by using the complete set of observed fp's in the cameras FOV. This results in an over-determined set of constraints for Δx_t which, in the absence of further statistical information, can be optimally used in a least squares sense as

$$\begin{bmatrix} f_{1,x}^t - f_{1,x}^{t-1} \\ \vdots \\ f_{K,x}^t - f_{K,x}^{t-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} s\Delta x_T^t \end{bmatrix} \qquad (17)$$

The fp constraint relations are expressed in this rather cumbersome way to establish the notation that we will require shortly when we have more than one ego-motion variable. Note that we have sets of over-determined equations of the form of $\mathbf{M} = \mathbf{A}\mathbf{P}$ where M denotes the measurement vector, A the model matrix and P the parameter vector of the unknowns. The least squares solution is $\mathbf{P} = \left(\mathbf{A}^T\mathbf{A}\right)^T \mathbf{A}^T\mathbf{M}$. This approach will be used for all of the ego-motion calculations. In this case, Δx becomes the average of the data contained in

$$\mathbf{M} = \begin{bmatrix} f_{1,x}^t - f_{1,x}^{t-1} \\ \vdots \\ f_{K,x}^t - f_{K,x}^{t-1} \end{bmatrix} \tag{18}$$

Next consider the more general motion of a translation in both the $\mathbf{x_w}$ and $\mathbf{y_w}$ directions from the frame t-1 to the current frame t . Use the notation that we have developed. We now have $\mathbf{T}_t = \begin{bmatrix} x_T^t & y_T^t & -h \end{bmatrix}^T$ such that the affine transformation is given as

$$\mathbf{A_t} = s\begin{bmatrix} 1 & 0 & -x_T^t \\ 0 & 1 & -y_T^t \end{bmatrix} \tag{19}$$

Again assume that the sets of feature points of $f_{c,k}^t$ for the current frame and $f_{c,k}^{t-1}$ from the previous frame are available. The differential change $\Delta \mathbf{A_t} \ \square \ \mathbf{A_t} - \mathbf{A_{t-1}}$ is now

$$\Delta \mathbf{A_t} = s\begin{bmatrix} 0 & 0 & -\Delta x_T^t \\ 0 & 0 & -\Delta y_T^t \end{bmatrix} \tag{20}$$

where $\Delta x_T^t \ \square \ x_T^t - x_T^{t-1}$ and $\Delta y_T^t \ \square \ y_T^t - y_T^{t-1}$. Extending the notation developed for the 1D case

$$\begin{bmatrix} f_{k,x}^t - f_{k,x}^{t-1} \\ f_{k,y}^t - f_{k,y}^{t-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} s\Delta x_T^t \\ s\Delta y_T^t \end{bmatrix} \tag{21}$$

This is of course separable into two independent sets of constraints over the K fp's as

$$
\begin{bmatrix} f_{1,x}^t - f_{1,x}^{t-1} \\ \vdots \\ f_{K,x}^t - f_{K,x}^{t-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} s\Delta x_T^t \end{bmatrix}
$$

$$
\begin{bmatrix} f_{1,y}^t - f_{1,y}^{t-1} \\ \vdots \\ f_{K,y}^t - f_{K,y}^{t-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} s\Delta y_T^t \end{bmatrix}
$$

(22)

These are solvable in the LS sense which in this case just amounts to determining the mean (over k) as

$$
-s\Delta x_t = mean\left(f_{k,x}^t - f_{k,x}^{t-1} \right)
$$
$$
-s\Delta y_t = mean\left(f_{k,y}^t - f_{k,y}^{t-1} \right)
$$

(23)

This provides the estimate of the differential motion of the camera based on two consecutive frames. If the requirement was essentially to only estimate the incremental translation of the camera between the two frames with no prior information then (17) would constitute the best estimate possible. However, we are generally interested in the translation motion over multiple frames. Extending the LS solution over multiple frames significantly improves the performance. For example if we make the additional assumption that the translation velocity is constant over three frames then we can extend the LS solution as

$$
\begin{bmatrix} f_{1,x}^t - f_{1,x}^{t-1} \\ \vdots \\ f_{K,x}^t - f_{K,x}^{t-1} \\ f_{1,x}^{t+1} - f_{1,x}^t \\ \vdots \\ f_{K,x}^{t+1} - f_{K,x}^t \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} s\Delta x_T^t \end{bmatrix}
$$

$$
\begin{bmatrix} f_{1,y}^t - f_{1,y}^{t-1} \\ \vdots \\ f_{K,y}^t - f_{K,y}^{t-1} \\ f_{1,y}^{t+1} - f_{1,y}^t \\ \vdots \\ f_{K,y}^{t+1} - f_{K,y}^t \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} s\Delta y_T^t \end{bmatrix}
$$

(24)

Note that generally this would result in a variance in the estimate of Δx_t and Δy_t that is half of the variance of using the single pair of frames. The message here is that if the camera

trajectory is smooth then a significant reduction of the variance of the ego-motion variable estimation is possible.

An example of 1Dmotion of a translating square subjected to noise. The first frame of the video is shown in Figure 5 together with the four tracked fp's as green donuts and the LKP trajectory of each fp shown in blue. White uniform noise is then added to each frame that is independent from frame to frame. The image is then smoothed which gives the noise a blotchy texture.

Figure 3-5 *Tracked square with fp's indicated and the LKP trajectories*

The plot in Figure 3-6 is the tracked fp's which is very clean as the motion in x is 3 pixels/frame which is an integer. Figure 3-7 shows the differentials of Δx_t and Δy_t. The velocity in x is changed to 2.5 pixels/frame. Figure 3-8 shows the differential of Δx_t which now has a ripple due to the video quantizing effect. To avoid this problem it is necessary to interpolate the video. Figure 9 shows Δx_t and Δy_t for vx=2.5 and vy=1.2 showing the quantization effects of the block pixel movement.
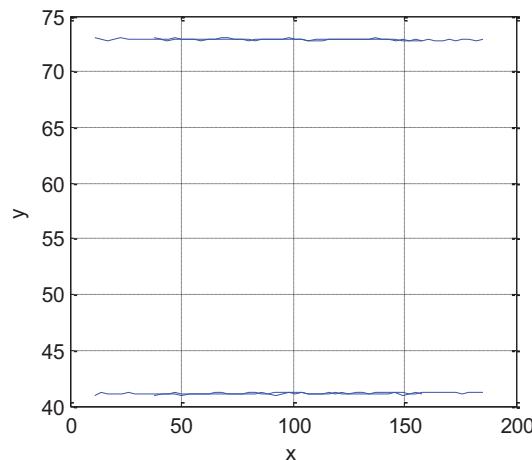
Figure 6 *Trajectories of fp's*
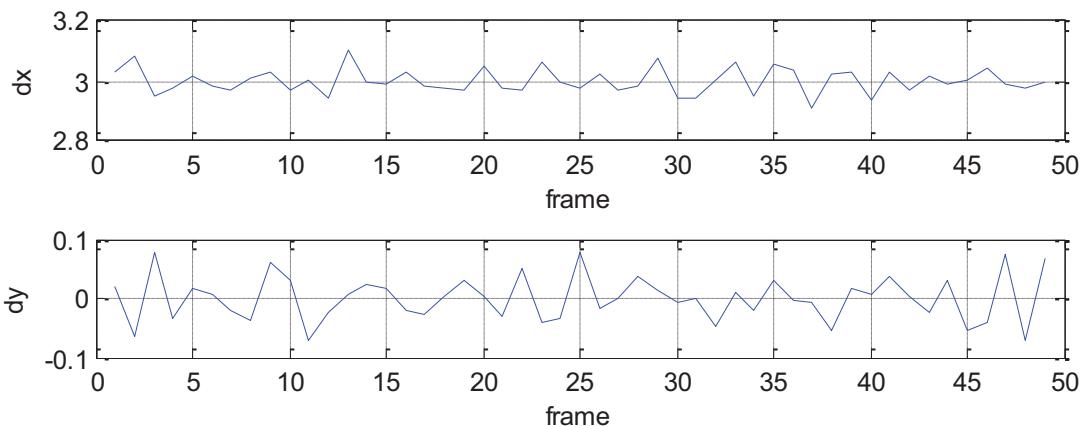
Figure 3-7 *Recovered differentials of Δx_t and Δy_t with vx=3*
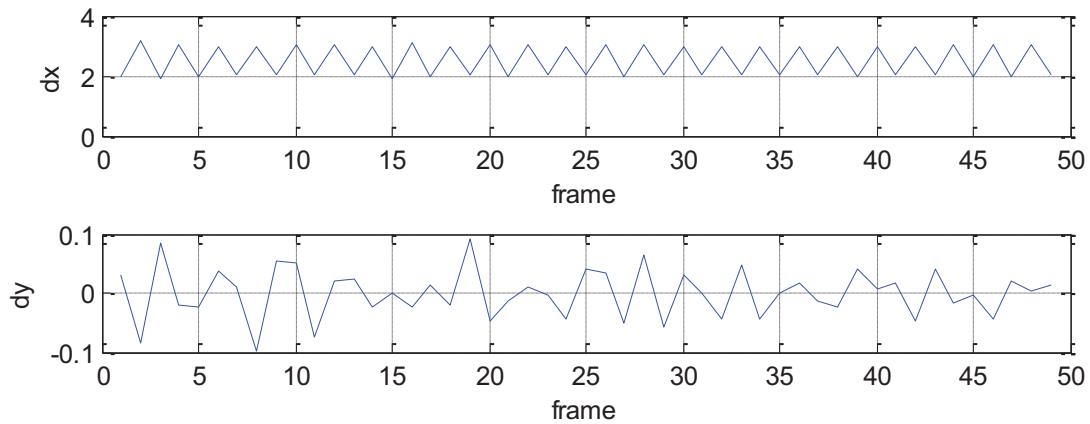
Figure 3-8 *Recovered differentials of Δx_t and Δy_t with vx=2.5*
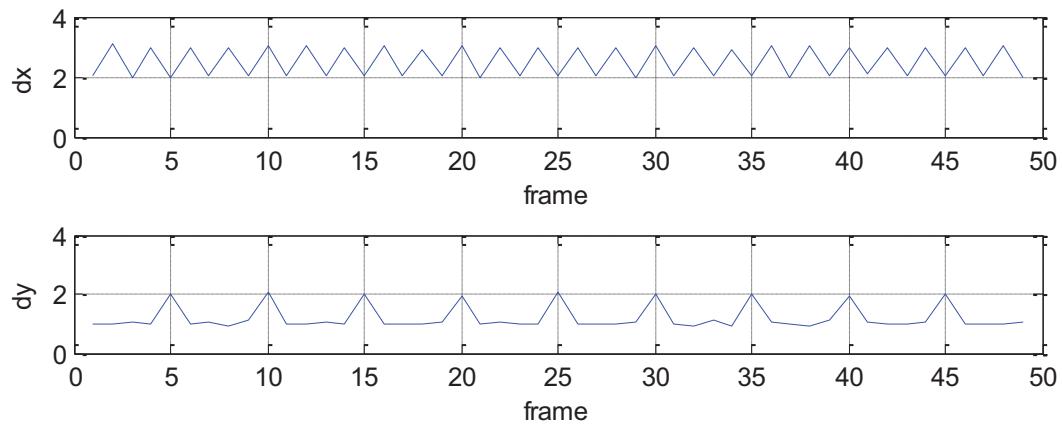
Figure 3-9 *Recovered differentials of Δx_t and Δy_t with vx=2.5 and vy=1.2*

3-4. Rotation of the World axis

Next consider the camera at a fixed position such that $x_T^t = y_T^t = 0$. As before the height is fixed at $z_T^t = -h$. The camera is rotated in a positive sense around the $\mathbf{z_W}$ axis of (a_z) as illustrated in Figure 10 such that the affine transformation is

$$
\mathbf{A_t} = s \begin{bmatrix} X_c \Box X_W & X_c \Box Y_W & 0 \\ Y_c \Box X_W & Y_c \Box Y_W & 0 \end{bmatrix} = s \begin{bmatrix} \cos(a_z) & -\sin(a_z) & 0 \\ \sin(a_z) & \cos(a_z) & 0 \end{bmatrix} \tag{25}
$$

Note that we define the sense of rotation relative to the right hand but we assume a left hand coordinate system for both the camera and the world as before.
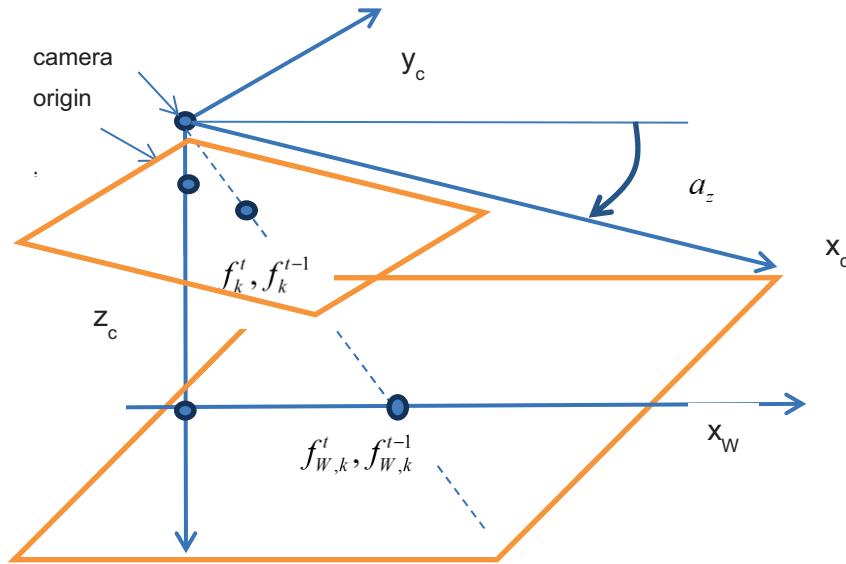
Figure 3-10 *Rotation of fp in camera FOV by* a_z

As before, we have

$$
\begin{bmatrix} f_{k,x}^t \\ f_{k,y}^t \end{bmatrix} = \mathbf{A_t} \begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \\ 1 \end{bmatrix} \tag{26}
$$

As there is no translation this transformation can be simplified to

$$\begin{bmatrix} f_{k,x}^t \\ f_{k,y}^t \end{bmatrix} = s \begin{bmatrix} \cos(a_z^t) & -\sin(a_z^t) \\ \sin(a_z^t) & \cos(a_z^t) \end{bmatrix} \begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \end{bmatrix}$$

(27)

where a_z^t is the azimuth rotation angle between the world and camera frame at time t which we cannot determine absolutely. However, we can determine relative changes as

$$\begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix} = \mathbf{A}_{t-1} \begin{bmatrix} f_{W,k,x} \\ f_{W,k,y} \end{bmatrix}$$

$$\begin{bmatrix} f_{k,x}^t \\ f_{k,y}^t \end{bmatrix} = \mathbf{A}_t \mathbf{A}_{t-1}^T \begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix}$$

$$= \begin{bmatrix} \cos(a_z^t) & -\sin(a_z^t) \\ \sin(a_z^t) & \cos(a_z^t) \end{bmatrix} \begin{bmatrix} \cos(a_z^{t-1}) & \sin(a_z^{t-1}) \\ -\sin(a_z^{t-1}) & \cos(a_z^{t-1}) \end{bmatrix} \begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix}$$

$$= \begin{bmatrix} \cos(a_z^t)\cos(a_z^{t-1}) + \sin(a_z^t)\sin(a_z^{t-1}) & \sin(a_z^t)\cos(a_z^{t-1}) - \cos(a_z^t)\sin(a_z^{t-1}) \\ -\sin(a_z^t)\cos(a_z^{t-1}) + \cos(a_z^t)\sin(a_z^{t-1}) & \cos(a_z^t)\cos(a_z^{t-1}) + \sin(a_z^t)\sin(a_z^{t-1}) \end{bmatrix} \begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix}$$

(28)

$$= \begin{bmatrix} \cos(\Delta a_z^t) & -\sin(\Delta a_z^t) \\ \sin(\Delta a_z^t) & \cos(\Delta a_z^t) \end{bmatrix} \begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix}$$

$$= \mathbf{R}_{\Delta t}^t \begin{bmatrix} f_{k,x}^{t-1} \\ f_{k,y}^{t-1} \end{bmatrix}$$

where

$$\Delta a_z^t = a_z^t - a_z^{t-1}$$

For notational convenience write

$$\begin{aligned} C_t &= \cos(\Delta a_z^t) \\ S_t &= \sin(\Delta a_z^t) \end{aligned}$$

(29)

such that

$$\mathbf{R}_{\Delta t}^t = \begin{bmatrix} C_t & -S_t \\ S_t & C_t \end{bmatrix}$$

(30)

and then $f_k^t = \mathbf{R}_{\Delta t}^t f_k^{t-1}$ specifically for the kth fp. Note that for pure differential rotation at a constant height the scaling factor cancels out. Combining all of the K fp's in a LS formulation to estimate Δa_z^t, it is most convenient to assume the redundant linear estimation of

$$
\begin{bmatrix} f_{1,x}^t \\ f_{1,y}^t \\ \vdots \\ f_{K,x}^t \\ f_{K,y}^t \end{bmatrix} = \begin{bmatrix} f_{1,x}^{t-1} & -f_{1,y}^{t-1} \\ f_{1,y}^{t-1} & f_{1,x}^{t-1} \\ \vdots & \vdots \\ f_{K,x}^{t-1} & -f_{K,y}^{t-1} \\ f_{K,y}^{t-1} & f_{K,x}^{t-1} \end{bmatrix} \begin{bmatrix} C_t \\ S_t \end{bmatrix}
\tag{31}
$$

The total accumulated rotation can be inferred from the rotation matrix of

$$
f_k^t = \prod_{p=1}^{t} \mathbf{R}_{\Delta t}^p \, f_k^0
\tag{32}
$$

Hence by multiplying the estimated differential rotation matrices we can determine the total azimuth rotation angle from the initial frame to the tth frame. This is approximately equivalent to the processing of

$$
\Delta a_{z,t} = \mathrm{atan2}\left(C_t, S_t\right)
$$
$$
a_{z,t} = \sum_{i=1}^{t} \Delta a_{z,i}
\tag{33}
$$

In summary then, the K pairs of corresponding fp's $f_k^t \Leftrightarrow f_k^{t-1}$ are determined from which the linear 2K set of constraints are determined based on the parameters of $\{S_t, C_t\}$. The LS pseudo-inverse of the constraint matrix is determined resulting in the LS estimate of $\{S_t, C_t\}$ from which the differential azimuth rotation of $\Delta a_{z,t}$ is determined. From this the accumulated azimuth rotation from the first image is determined as $a_{z,t}$.

3-5. General affine transformations

Next consider that the camera is rotated and translated simultaneously by arbitrary differential amounts. This is represented by the affine transformation of the motion of the kth feature point from the $(t-1)^{th}$ to the tth frame is described by the affine transformation. Instead of stating the

affine transformation directly we start with the perspective transformation. Begin by defining the perspective transformation between the static world fp's as

$$
\begin{aligned}
f^{t-1} &= \mathbf{R}_{t-1}\left(f^W - \mathbf{T}_{t-1}\right) \\
f^t &= \mathbf{R}_t\left(f^W - \mathbf{T}_t\right)
\end{aligned}
\tag{34}
$$

Then we have

$$
f^W = \mathbf{R}_{t-1}^T f^{t-1} + \mathbf{T}_{t-1}
\tag{35}
$$

Then

$$
\begin{aligned}
f^t &= \mathbf{R}_t\left(\mathbf{R}_{t-1}^T f^{t-1} + \mathbf{T}_{t-1} - \mathbf{T}_t\right) \\
&= \mathbf{R}_t\mathbf{R}_{t-1}^T\left(f^{t-1} - \mathbf{R}_{t-1}\left(\mathbf{T}_t - \mathbf{T}_{t-1}\right)\right)
\end{aligned}
\tag{36}
$$

We have

$$
\mathbf{R}_t\mathbf{R}_{t-1}^T = \begin{bmatrix}
\mathbf{X}_c^t\square\mathbf{X}_c^{t\text{-}1} & \mathbf{X}_c^t\square\mathbf{Y}_c^{t\text{-}1} & \mathbf{X}_c^t\square\mathbf{Z}_c^{t\text{-}1} \\
\mathbf{Y}_c^t\square\mathbf{X}_c^{t\text{-}1} & \mathbf{Y}_c^t\square\mathbf{Y}_c^{t\text{-}1} & \mathbf{Y}_c^t\square\mathbf{Z}_c^{t\text{-}1} \\
\mathbf{Z}_c^t\square\mathbf{X}_c^{t\text{-}1} & \mathbf{Z}_c^t\square\mathbf{Y}_c^{t\text{-}1} & \mathbf{Z}_c^t\square\mathbf{Z}_c^{t\text{-}1}
\end{bmatrix} = \begin{bmatrix}
\cos(\Delta az) & \sin(\Delta az) & 0 \\
-\sin(\Delta az) & \cos(\Delta az) & 0 \\
0 & 0 & 1
\end{bmatrix}
\tag{37}
$$

which is the incremental rotation from the camera coordinate in the t-1 frame to the world frame and then back to the camera coordinate in the tth frame. $\left(\mathbf{T}_t - \mathbf{T}_{t-1}\right)$ is the incremental translation between the t-1 and t frames relative to the world coordinates. As $\mathbf{R}_{t-1}$ rotates this vector into the camera frame at the $(t\text{-}1)^{\text{th}}$ frame, $\mathbf{R}_{t-1}\left(\mathbf{T}_t - \mathbf{T}_{t-1}\right)$ is the incremental translation vector relative to the $(t\text{-}1)^{\text{th}}$ frame. Hence

$$
\mathbf{R}_t\mathbf{R}_{t-1}^T\mathbf{R}_{t-1}\left(\mathbf{T}_t - \mathbf{T}_{t-1}\right) = \mathbf{R}_t\left(\mathbf{T}_t - \mathbf{T}_{t-1}\right)
\tag{38}
$$

is the translation vector relative to the tth frame. Hence

$$
\mathbf{R}_t\left(\mathbf{T}_t - \mathbf{T}_{t-1}\right) = \begin{bmatrix}
-\Delta x_t \\
-\Delta y_t \\
0
\end{bmatrix}
\tag{39}
$$

Now the perspective transformation

$$
\begin{bmatrix}
\left[\mathbf{R}_t\mathbf{R}_{t-1}^T\right]_{11} & \left[\mathbf{R}_t\mathbf{R}_{t-1}^T\right]_{12} & \left[\mathbf{R}_t\left(\mathbf{T}_t-\mathbf{T}_{t-1}\right)\right]_1 \\
\left[\mathbf{R}_t\mathbf{R}_{t-1}^T\right]_{21} & \left[\mathbf{R}_t\mathbf{R}_{t-1}^T\right]_{22} & \left[\mathbf{R}_t\left(\mathbf{T}_t-\mathbf{T}_{t-1}\right)\right]_2 \\
\left[\mathbf{R}_t\mathbf{R}_{t-1}^T\right]_{31} & \left[\mathbf{R}_t\mathbf{R}_{t-1}^T\right]_{32} & \left[\mathbf{R}_t\left(\mathbf{T}_t-\mathbf{T}_{t-1}\right)\right]_3
\end{bmatrix}
=
\begin{bmatrix}
\cos(\Delta az) & \sin(\Delta az) & \Delta x_t \\
-\sin(\Delta az) & \cos(\Delta az) & \Delta y_t \\
0 & 0 & 0
\end{bmatrix}
$$

From this the incremental affine transformation applied to the fp's becomes

$$
\begin{bmatrix} f_{x,k}^t \\ f_{y,k}^t \\ 1 \end{bmatrix}
=
\begin{bmatrix}
C_t & S_t & \Delta x_t \\
-S_t & C_t & \Delta y_t \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} f_{x,k}^{t-1} \\ f_{y,k}^{t-1} \\ 1 \end{bmatrix}
\tag{40}
$$

From which the differential affine transformation is defined as

$$
\mathbf{\Delta A}_t =
\begin{bmatrix}
C_t & S_t & \Delta x_t \\
-S_t & C_t & \Delta y_t \\
0 & 0 & 1
\end{bmatrix}
\tag{41}
$$

Now consider the cascaded transformations of the first two frames. We have

$$
\mathbf{\Delta A}_2\mathbf{\Delta A}_1 =
\begin{bmatrix}
C_2 & S_2 & \Delta x_2 \\
-S_2 & C_2 & \Delta y_2 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
C_1 & S_1 & \Delta x_1 \\
-S_1 & C_1 & \Delta y_1 \\
0 & 0 & 1
\end{bmatrix}
$$

$$
=
\begin{bmatrix}
C_2C_1 - S_2S_1 & S_1C_1 + S_2C_2 & C_2\Delta x_1 + S_2\Delta y_1 + \Delta x_2 \\
-S_1C_1 - S_2C_2 & C_2C_1 - S_2S_1 & -S_2\Delta x_1 + C_2\Delta y_1 + \Delta y_2 \\
0 & 0 & 1
\end{bmatrix}
\tag{42}
$$

Note that

$$
C_2C_1 - S_2S_1 = \cos\left(\Delta az_1 + \Delta az_2\right)
$$
$$
S_1C_1 + S_2C_2 = \sin\left(\Delta az_1 + \Delta az_2\right)
$$
$$
\begin{bmatrix} C_2\Delta x_1 + S_2\Delta y_1 + \Delta x_2 \\ -S_2\Delta x_1 + C_2\Delta y_1 + \Delta y_2 \end{bmatrix}
=
\begin{bmatrix} C_2 & S_2 \\ -S_2 & C_2 \end{bmatrix}
\begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix}
+
\begin{bmatrix} \Delta x_2 \\ \Delta y_2 \end{bmatrix}
\tag{43}
$$

From this we can define the accumulated affine transformation as

$$
\mathbf{A}_t =
\begin{bmatrix}
\cos(az_t) & \sin(az_t) & x_t \\
-\sin(az_t) & \cos(az_t) & y_t \\
0 & 0 & 1
\end{bmatrix}
\tag{44}
$$

with the components related to the rotation and translation of the camera relative to the world reference at the tth frame such that

$$\mathbf{R}_t = \begin{bmatrix} \cos(az_t) & \sin(az_t) & 0 \\ -\sin(az_t) & \cos(az_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}_t = \begin{bmatrix} -x_t \\ -y_t \\ -h \end{bmatrix} \tag{45}$$

From this we have the accumulated affine transformation given as a recursive relation as

$$\mathbf{A}_t = \Delta\mathbf{A}_t\mathbf{A}_{t-1} \tag{46}$$

There are two approaches to estimating the differential affine transformation coefficients. The first is to consider the 2K constraints based on

$$\begin{bmatrix} f_{x,k}^t \\ f_{y,k}^t \\ 1 \end{bmatrix} = \begin{bmatrix} C_t & S_t & \Delta x_t \\ -S_t & C_t & \Delta y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{x,k}^{t-1} \\ f_{y,k}^{t-1} \\ 1 \end{bmatrix} \tag{47}$$

That is we form for each fp a pair of constraints as

$$\begin{bmatrix} f_{x,k}^t \\ f_{y,k}^t \end{bmatrix} = \begin{bmatrix} f_{x,k}^{t-1} & f_{y,k}^{t-1} & 1 & 0 \\ f_{y,k}^{t-1} & -f_{x,k}^{t-1} & 0 & 1 \end{bmatrix} \begin{bmatrix} C_t \\ S_t \\ \Delta x_t \\ \Delta y_t \end{bmatrix} \tag{48}$$

Due to changes in camera height, distortions and sources of measurement noise, in practice $C_t^2 + S_t^2 \neq 1$. This discrepancy is removed by the atan2() processing in regards to determining the angle change. The deviation of $C_t^2 + S_t^2$ from 1 provides an estimate of the change in height relative to the current height.

As an alternative to the LS solution of $\{C_t, S_t, \Delta x_t, \Delta y_t\}$ we can solve for the six affine coefficients directly as given by the fp update of

$$\begin{bmatrix} f_{x,k}^t \\ f_{y,k}^t \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{x,k}^{t-1} \\ f_{y,k}^{t-1} \\ 1 \end{bmatrix} \tag{49}$$

Note that there are 6 unknowns whereas previously there were 4 unknowns. However, the extra degrees of freedom are not meaningful as there are additional constraints implied by the

$\{C_t, S_t, \Delta x_t, \Delta y_t\}$ system. Note that the general affine transformation can accommodate rotation about any arbitrary center point whereas the $\{C_t, S_t, \Delta x_t, \Delta y_t\}$ system assumes rotation to be fixed about the camera center. Also the scaling is the same in the $\mathbf{x}_e$ and $\mathbf{y}_e$ directions which accounts for the additional DOF. To use this system it is necessary to account for the rotation center and the different scaling factors when mapping back to the ground reference system. The coefficients of $\{a, b, c, d, e, f\}$ are solved for with two sets of equations as in a previous chapter as

$$
\begin{bmatrix} f_{ex,1}^{t} \\ f_{ex,2}^{t} \\ \vdots \\ f_{ex,N}^{t} \end{bmatrix} = \begin{bmatrix} f_{ex,1}^{t-1} & f_{ey,1}^{t-1} & 1 \\ f_{ex,2}^{t-1} & f_{ey,2}^{t-1} & 1 \\ \vdots & \vdots & \vdots \\ f_{ex,N}^{t-1} & f_{ey,N}^{t-1} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}
\tag{50}
$$

$$
\begin{bmatrix} f_{ey,1}^{t} \\ f_{ey,2}^{t} \\ \vdots \\ f_{ey,N}^{t} \end{bmatrix} = \begin{bmatrix} f_{ex,1}^{t-1} & f_{ey,1}^{t-1} & 1 \\ f_{ex,2}^{t-1} & f_{ey,2}^{t-1} & 1 \\ \vdots & \vdots & \vdots \\ f_{ex,N}^{t-1} & f_{ey,N}^{t-1} & 1 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \end{bmatrix}
\tag{51}
$$

3.5 3D Perspective ego-motion based on a 2D marker and unknown fp`s

Next consider that the camera undergoes 6 DOF motion change while viewing a known 2D poster marker on a wall as shown in Figure 11. Here the marker is in the $Z_w = 0$ plane at a known location. Next consider the perspective mapping of a 2D pattern on the $Z_w = 0$ plane in the world coordinates that is imaged in the camera by projection onto the image plane. Another illustration is given in Figure 3-12 where a set of fp's is used to provide continuous 3D egomotion. The concept is that there will always be several fp's in the camera FOV which are sequentially self calibrating. That is as a new fp enters the FOV the existing fp's will be used to calibrate the position of the new fp's. This will give us the perspective coefficients which are related to $\{\mathbf{R}, \mathbf{T}\}$. As $Z_w = 0$ we can write

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} [\mathbf{R_1}]_1 & [\mathbf{R_1}]_2 \\ [\mathbf{R_2}]_1 & [\mathbf{R_2}]_2 & -\mathbf{RT} \\ [\mathbf{R_3}]_1 & [\mathbf{R_3}]_2 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \qquad (52)$$

where $\mathbf{H}$ is a 3x3 matrix with elements defined as

$$\mathbf{H} = \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix} \qquad (53)$$
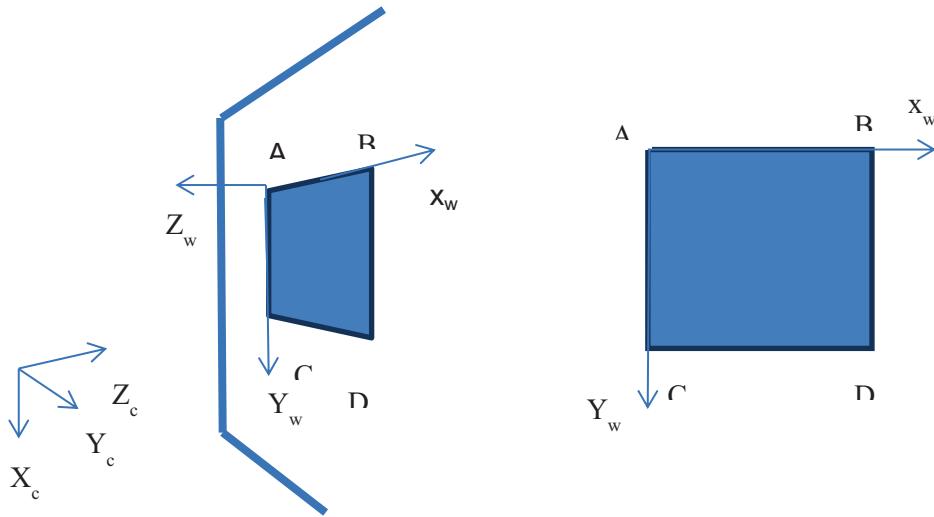
Figure 3-11 *Square poster viewed from a camera in the image plane can be restored to the undistorted square shape based on the perspective transformation.*
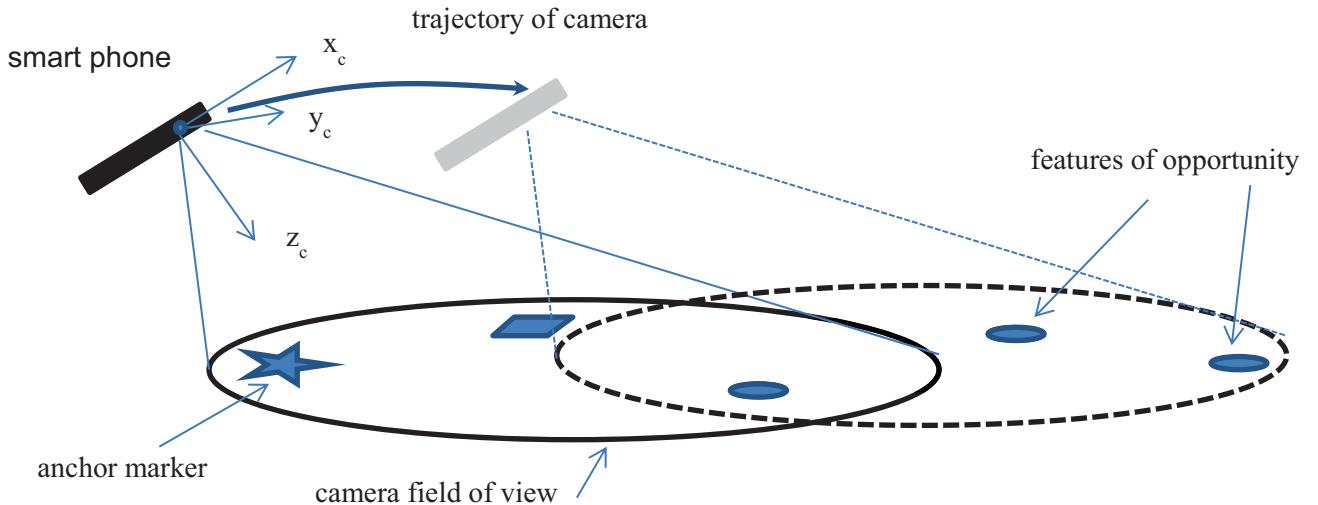
Figure 3-12 Concept of 3D ego-motion based on viewing the floor surface with various fp's in the current camera FOV

Note that the first two columns of $\mathbf{H}$ are the first two columns of $\mathbf{R}$ and the third column of $\mathbf{H}$ is $-\mathbf{RT}$ as before. Every fp of the marker that is distinguishable provides two constraints from which the 9 components of $\mathbf{H}$ can be determined. We have

$$
\begin{aligned}
x = X_c/Z_c &= \frac{H_{11}X_w + H_{12}Y_w + H_{13}}{H_{31}X_w + H_{32}Y_w + H_{33}} \\
y = Y_c/Z_c &= \frac{H_{21}X_w + H_{22}Y_w + H_{23}}{H_{31}X_w + H_{32}Y_w + H_{33}}
\end{aligned}
\tag{54}
$$

which is rearranged as

$$
\begin{aligned}
-H_{11}X_w - H_{12}Y_w - H_{13} + H_{31}xX_w + H_{32}xY_w + H_{33}x = 0 \\
-H_{21}X_w - H_{22}Y_w - H_{23} + H_{31}yX_w + H_{32}yY_w + H_{33}y = 0
\end{aligned}
\tag{55}
$$

This results in a pair of constraints expressed as

$$\mathbf{u}_x\mathbf{b} = 0$$
$$\mathbf{u}_y\mathbf{b} = 0 \tag{56}$$

where

$$\mathbf{b} = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{21} & H_{22} & H_{23} & H_{31} & H_{32} & H_{33} \end{bmatrix}^T$$
$$\mathbf{u}_x = \begin{bmatrix} -X_w & -Y_w & -1 & 0 & 0 & 0 & xX_w & xY_w & x \end{bmatrix} \tag{57}$$
$$\mathbf{u}_y = \begin{bmatrix} 0 & 0 & 0 & -X_w & -Y_w & -1 & yX_w & yY_w & y \end{bmatrix}$$

For a set of K fp's that can be identified from the marker we have

$$\begin{bmatrix} \mathbf{u}_{x,1} \\ \mathbf{u}_{y,1} \\ \vdots \\ \mathbf{u}_{x,K} \\ \mathbf{u}_{y,K} \end{bmatrix} \mathbf{b} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{58}$$

We then apply the singular value decomposition (svd) solution to determine the components of $\mathbf{b}$ based on using the right singular vector corresponding to the smallest singular value. Once the components of $\mathbf{H}$ is determined then we can map these into the set of variables $\{a_x, a_y, a_z, x_T, y_T, z_T, s\}$ from which the $\{\mathbf{R}, \mathbf{T}\}$ matrices can be determined. The additional variable of s is a scaling factor that is necessary as $\mathbf{H}$ will generally have an arbitrary scaling associated with it. Hence s is a nuisance parameter but necessary in order to make the mapping of $\mathbf{H} \rightarrow \{a_x, a_y, a_z, x_T, y_T, z_T\}$ possible. The following Matlab routine shows the procedure of extracting $\{a_x, a_y, a_z, x_T, y_T, z_T\}$ from $\mathbf{H}$.

First we generate an $\mathbf{H}$ matrix to work with.

```
% Hmat
function H = Hmat(ax,ay,az,T,s)

% generate the RT matrix
Cx = cos(ax);Sx = sin(ax);
Cy = cos(ay);Sy = sin(ay);
Cz = cos(az);Sz = sin(az);

Rx = [[1,0,0];[0,Cx,Sx];[0,-Sx,Cx]];
Ry = [[Cy 0 Sy];[0 1 0];[-Sy 0 Cy]];
```

```
Rz = [[Cz Sz 0];[-Sx Cz 0];[0 0 1]];
R = Rz*Ry*Rx;
H = s*[R(:,1:2),-R*T];
```

The function to determine the error vector used in the nonlinear equation solver is given as

```
function F=H2RTfun(H,x)
ax = x(1);
ay = x(2);
az = x(3);
T = x(4:6);
s = x(7);
e = H-Hmat(ax,ay,az,T,s);
F=zeros(9,1);
F(:) = e;
```

Finally the call to fsolve(), Matlab's generic equation solver is

```
options=optimset('Display','iter');   % Option to display output
[aTc,fval,exitflag,output,jacobian] =  fsolve(@(x)H2RTfun(H,x),aT,options);
```

The fsolve() routine has no issue with quickly converging on the correct values of the parameters.

Next consider the more general problem of the markers unknown to the camera. However, to simplify the scenario it will be assumed that the perspective transformation is known at the initial point where the camera begins it's trajectory. At this point it then begins to observe unknown markers. It is also assumed that the initial known template and fp's are all on the $Z_w = 0$ plane.

Consider that the camera undergoes an arbitrary translation and rotation relative to the world reference such that at the time that the tth frame is available from the camera the perspective rotation matrix and translation vector are denoted as $\{\mathbf{R}_t, \mathbf{T}_t\}$. The objective is that given $\{\mathbf{R}_{t-1}, \mathbf{T}_{t-1}\}$ for the (t-1)th frame then determine $\{\mathbf{R}_t, \mathbf{T}_t\}$ for the tth frame based on the movement of the fp's in the image plane as recorded by the camera. As derived before, the transformation between the two camera positions is given as

$$\mathbf{P}_c^t = \mathbf{R}_t \mathbf{R}_{t-1}^T \left(\mathbf{P}_c^{t-1} - \mathbf{R}_{t-1} \left(\mathbf{T}_t - \mathbf{T}_{t-1} \right) \right) \tag{59}$$

where $\mathbf{P}_c^{t-1}$ and $\mathbf{P}_c^t$ are the position vectors of the fp in the camera reference in the $(t-1)^{th}$ and t^{th} frame.   Define

$$\mathbf{R}_\Delta^t = \mathbf{R}_t \mathbf{R}_{t-1}^T \tag{60}$$

as the rotation matrix between the two camera coordinates from frame t-1 to frame t. That is

$$\mathbf{R}_\Delta^t = \begin{bmatrix} \mathbf{X}_c^t \square \mathbf{X}_c^{t-1} & \mathbf{X}_c^t \square \mathbf{Y}_c^{t-1} & \mathbf{X}_c^t \square \mathbf{Z}_c^{t-1} \\ \mathbf{Y}_c^t \square \mathbf{X}_c^{t-1} & \mathbf{Y}_c^t \square \mathbf{Y}_c^{t-1} & \mathbf{Y}_c^t \square \mathbf{Z}_c^{t-1} \\ \mathbf{Z}_c^t \square \mathbf{X}_c^{t-1} & \mathbf{Z}_c^t \square \mathbf{Y}_c^{t-1} & \mathbf{Z}_c^t \square \mathbf{Z}_c^{t-1} \end{bmatrix} \tag{61}$$

Also define

$$\mathbf{T}_\Delta^t = \mathbf{R}_{t-1} \left(\mathbf{T}_t - \mathbf{T}_{t-1} \right) \tag{62}$$

as the translation vector between the two camera positions relative to the camera frame at t-1. Then the transformation is given as

$$\mathbf{P}_c^t = \mathbf{R}_\Delta^t \left(\mathbf{P}_c^{t-1} - \mathbf{T}_\Delta^t \right) = \begin{bmatrix} \mathbf{R}_\Delta^t & -\mathbf{R}_\Delta^t \mathbf{T}_\Delta^t \end{bmatrix} \begin{bmatrix} \mathbf{P}_c^{t-1} \\ 1 \end{bmatrix} \tag{63}$$

Note that $\mathbf{P}_c^{t-1}$ is known as $\{\mathbf{R}_{t-1}, \mathbf{T}_{t-1}\}$ as well as the image coordinates of the fp from the $(t-1)^{th}$ frame are known. We can demonstrate this as we have the following equations

$$\begin{bmatrix} P_{c,x}^{t-1} \\ P_{c,y}^{t-1} \\ P_{c,z}^{t-1} \end{bmatrix} = \mathbf{R}_{t-1} \left(\begin{bmatrix} P_{W,x} \\ P_{W,y} \\ 0 \end{bmatrix} - \mathbf{T}_{t-1} \right)$$

$$x^{t-1} = s P_{c,x}^{t-1} / P_{c,z}^{t-1}$$

$$y^{t-1} = s P_{c,y}^{t-1} / P_{c,z}^{t-1} \tag{64}$$

which represents 5 equations and 5 unknowns as $\{ P_{c,x}^{t-1}, P_{c,y}^{t-1}, P_{c,z}^{t-1}, P_{W,x}, P_{W,y} \}$.   s is a known scaling factor.

Define $\mathbf{H}$ as a 3x4 matrix as

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}_\Delta^t & -\mathbf{R}_\Delta^t \mathbf{T}_\Delta^t \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ h_{3,1} & h_{3,2} & h_{3,3} & h_{3,4} \end{bmatrix} \tag{65}$$

The non-homogeneous image coordinates of the tth frame are

$$x_c^t = s \frac{\mathbf{P}_{c,x}^t}{\mathbf{P}_{c,z}^t}$$

$$y_c^t = s \frac{\mathbf{P}_{c,y}^t}{\mathbf{P}_{c,z}^t} \tag{66}$$

Let $h_i = \begin{bmatrix} h_{i,1} & h_{i,2} & h_{i,3} & h_{i,4} \end{bmatrix}$ denote the ith row of $\mathbf{H}$ and let $f = \begin{bmatrix} \mathbf{P}_{c,x}^{t-1} & \mathbf{P}_{c,y}^{t-1} & \mathbf{P}_{c,z}^{t-1} & 1 \end{bmatrix}$ denote the homogeneous fp coordinates such that

$$h_3 f x_c^t = h_1 f$$

$$h_3 f y_c^t = h_2 f \tag{67}$$

This results in a pair of homogeneous equations for each fp expressed as

$$\begin{bmatrix} f & 0 & 0 & 0 & 0 & -f x_c^t \\ 0 & 0 & 0 & 0 & f & -f y_c^t \end{bmatrix} \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{68}$$

From this a pair of constraints are formed for each fp correspondence which can be expressed as a homogeneous solution of the unknown values of h_{ij}. First a 1x12vector of h_{ij} denoted as h is defined as

$$h = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}^T$$

$$= \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{21} & h_{22} & h_{23} & h_{24} & h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix}^T \tag{69}$$

Set up the matrix component for the kth fp as

$$\begin{bmatrix} f_k & 0 & 0 & 0 & 0 & -f_k x_{c,k}^t \\ 0 & 0 & 0 & 0 & f_k & -f_k y_{c,k}^t \end{bmatrix} = \begin{bmatrix} a_{x,k}^t \\ a_{x,k}^t \end{bmatrix} \tag{70}$$

where $f_k = \begin{bmatrix} \mathbf{P}^{t-1}_{c,k,x} & \mathbf{P}^{t-1}_{c,k,y} & \mathbf{P}^{t-1}_{c,k,z} & 1 \end{bmatrix}$.

Note we start with the assumption that we have $\mathbf{P}^{t-1}_{c,k}$. Then from the corresponding fp's that we observe in frame t we can solve for the matrix $\mathbf{H}$. From the matrix $\mathbf{H}$ we get $\mathbf{R}^t_\Delta$ and $\mathbf{T}^t_\Delta$. Then we can solve for the rotation matrix as

$$\mathbf{R}_t = \mathbf{R}^t_\Delta \mathbf{R}_{t-1} \tag{71}$$

and

$$\mathbf{T}_t = \mathbf{R}^T_{t-1} \mathbf{T}^t_\Delta + \mathbf{T}_{t-1} \tag{72}$$

Then we have

$$\mathbf{P}^t_c = \mathbf{R}^t_\Delta \left(\mathbf{P}^{t-1}_c - \mathbf{T}^t_\Delta \right) \tag{73}$$

and we are set for another iteration.

The overall algorithm can be described as follows. First the known marker is observed in the camera at the known location for which we fix the world reference. From this the initial $\{\mathbf{R}_0, \mathbf{T}_0\}$ is established. Next the camera also observes some fp's of opportunity at unknown location except for the assumption that the fp's are located also on the $Z_w = 0$ plane. Through multiple consecutive frames, while both the known marker and fp's are in the camera FOV, the location of the fp's is determined in the world reference in the $Z_w = 0$ plane. As the camera is moved, the known marker will eventually drift out of the FOV and we have only the fp's of opportunity. However, as the location of these have been established, then we can determine the perspective transformation to the camera and also $\{\mathbf{R}_t, \mathbf{T}_t\}$. As the trajectory progresses, new fp's are going to enter the FOV with positions estimated based on the older set of fp's and hence the process of extracting $\{\mathbf{R}_t, \mathbf{T}_t\}$ can continue indefinitely as long as the fp's are all on the $Z_w = 0$ plane.

In a practical scenario there will be limitations and complications with this method. Due to noise and lens distortion of the camera there will be a drift issue. That is as the trajectory progresses and new fp's are added to the mix then the errors will accumulate. Also the camera will pick up fp's that are not on the $Z_w = 0$ plane. These will have to be processed based on determining the plane in the world reference where they reside.

3.7 Alternative 3D Method

While the previous method works in general 3D as will be shown experimentally, it is subject to drift as it is necessary to base $\{\mathbf{R}_t, \mathbf{T}_t\}$ on the previous step of $\{\mathbf{R}_{t-1}, \mathbf{T}_{t-1}\}$. Additionally it is necessary that the initial pose of the camera is known as eventually $\mathbf{R}_0$ is assumed to be known. Generally 3D egomotion of this type is based on epipolar geometry. However, to determine the essential matrix for the epipolar update, it is necessary to have fp's that are not on a plane. At the same time the powerful constraint of the fp's residing on the plane is not utilized. Hence another approach is considered which specifically takes advantage of the $Z_w = 0$ constraint. Consider that we have two images from two camera positions. For a given fp we have the position vector from camera 1 given as

$$X_1 = R_1 \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} + T_1$$

where R_1 is the rotation from the planar world to the camera in position 1 and T_1 is the translation from the world to camera 1 referenced to the camera coordinate. We have the image plane coordinates of the camera in position 1 such that the observable is

$$x_1 = \frac{s_c}{X_{1,z}} X_{1,x}$$

$$y_1 = \frac{s_c}{X_{1,z}} X_{1,y} \tag{28}$$

where s_c is the scaling factor. Likewise we have for camera position 2

$$X_2 = R_2 \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} + T_2$$

where R_2 is the rotation from the planar world to the camera in position 2 and T_2 is the translation from the world to camera 2 referenced to the camera 2 coordinates. Image coordinates are given as

$$x_2 = \frac{s_c}{X_{2,z}} X_{2,x}$$

$$y_2 = \frac{s_c}{X_{2,z}} X_{2,y}$$

(29)

Next consider the transformation of camera in position 2 relative to camera position 1 given as

$$X_2 = R_r X_1 + T_r$$

$$= R_r R_1 \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} + R_r T_1 + T_r$$

(30)

R_r is the rotation from camera frame 1 to camera frame 2 in frame 2. T_r is the translation from camera 1 to camera 2 in the frame of camera 2. That is relation

$$[R_r]_{i,j} = X_{2,i}^T X_{1,j}$$

(31)

where $X_{k,i}$ represents the ith basis vector of camera position k. Also

$$T_r = o_1 - o_2$$

where o_k is the camera center of the kth position relative to the camera position 2 frame. Now we define d is the perpendicular distance from the plane to o_1 (camera position 1) and N is the normal to the plane of $Z_W = 0$ relative to camera frame 1 such that $d = N^T X_1$. Note that there is an ambiguity as the normal to the world plane N can have two ambiguous directions. This is resolved by stipulating that $d > 0$. A trick is to set $1 = N^T X_1 / d$ such that

$$X_2 = R_r X_1 + 1 \cdot T_r$$

$$= R_r X_1 + \left(\frac{N^T X_1}{d} \right) T_r$$

$$= R_r X_1 + T_r \left(\frac{N^T X_1}{d} \right)$$

(32)

$$= \left(R_r + \frac{T_r N^T}{d} \right) X_1 = H X_1$$

(Note the order of the vectors can be changed as they are dotted to a scalar.) Then we have a matrix that relates these two positions as

$$X_2 = HX_1$$

$$H \square R_r + \frac{1}{d} T_r N^T \tag{33}$$

As observed, we have the ratio T/d which is dimensionless. Hence we cannot determine both the height of the camera and the translation simultaneously but only the ratio. Next scale the position vectors by normalizing to form the 2D homogeneous vectors as

$$\mathbf{x_1} = \lambda_1 X_1 = \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\mathbf{x_2} = \lambda_2 X_2 = \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{34}$$

We can then write $X_2 = HX_1$ relating the 2D homogeneous vectors as

$$\mathbf{x_2} = \lambda H \mathbf{x_1} \tag{35}$$

Note that the scaling constant λ is irrelevant as it cannot be determined anyways since H will only be known to within a scaling constant. Now consider that $\mathbf{x_2} \times \mathbf{x_2} = \mathbf{x_2}\mathbf{x_2} = 0$ such that

$$\mathbf{x_2} H \mathbf{x_1} = 0 \tag{36}$$

which is a set of homogeneous relations. Note that $\mathbf{x_2}$ can be expressed as a skew symmetric matrix operator as

$$\mathbf{x} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

To solve for the elements of H within a scaling constant the following procedure is used

$$a^j = x_1^j \otimes x_2^j$$

$$A = \begin{bmatrix} a^1 & \cdots & a^K \end{bmatrix}^T \tag{37}$$

Solve the SVD of

$$AH^s = 0 \tag{38}$$

where H^s is the stacked version of H corresponding to the right singular vector with the smallest singular value. The second singular value of H should be 1 which gives a way of normalizing H. After determining H then $H = R_r + \dfrac{1}{d}T_r N^T$ determine R_r and $\dfrac{1}{d}T_r$. Note that we can only get a scaled version of T_r/d such that the height is required or some measure of distance for calibration. Another complication is that N is in the camera position 1 frame.

Example - Consider the simplest example of a simple translation of t_x to the right where the plane of the fp's is a distance d from the camera positions and the normal to the plane is $N = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. (Note that $N \neq \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$ as $N^T X_1 = d$). We have $T_r = \begin{bmatrix} -t_x & 0 & 0 \end{bmatrix}^T$ and $R_r = I_3$. Hence,

$$H = R_r + \frac{1}{d}T_r N^T$$

$$= \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} + \begin{bmatrix} -t_x/d \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -t_x/d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From $\mathbf{x_2} = \lambda H \mathbf{x_1}$ we have

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} 1 & 0 & -t_x/d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} x_1 - t_x/d \\ y_1 \\ 1 \end{bmatrix}$$

which makes sense.

Next consider a simulation of determining H arbitrary rotation angles and translation. Again we consider a set of random feature points distributed in the world plane around the origin. Note

that care has to be taken that the translation vector is referenced to the correct coordinate system.

First define the parameters. d is the height of the camera 1 position, (tx,ty,tz) are the components of the translation vector in the coordinates of position 1, sc is the scaling of the camera to pixel index, and (ax,ay,az) are the rotation angles of position 2 relative to position 1. Provided that sc is the same in the x and y directions it has not impact on the estimation of H.

```
d = 10; % height of camera for position 1
tx = 1;ty=0.5;tz=0;  % relative motion between positions 1 and 2
ax = 0.1;ay = 0.2;az = 0.1;
sc = 1; % scaling of camera
K = 10; % number of random points
```

Next we generate the relative rotation and translation matrices denoted as Rr and Tr respectively.

```
Cx = cos(ax);Sx = sin(ax);
Cy = cos(ay);Sy = sin(ay);
Cz = cos(az);Sz = sin(az);
Rx = [[1,0,0];[0,Cx,Sx];[0,-Sx,Cx]];
Ry = [[Cy 0 Sy];[0 1 0];[-Sy 0 Cy]];
Rz = [[Cz Sz 0];[-Sz Cz 0];[0 0 1]];
Rr = Rz*Ry*Rx;
Tr = [tx;ty;tz];
```

The position 1 is given the translation of T_1 and R_1 relative to the position 1 frame

```
T_1=R_1 * [0;0;d];
R_1 =eye(3);
H_1 = [[sc 0 0];[0 sc 0];[0 0 1]] * [R_1(:,1:2),R_1*T_1];
```

The position 2 is given the translation of T_2 and R_2 relative to the position 2 frame

```
T_2 = R_2*(T_1 + Tr);
R_2 =Rr*R_1;
H_2 = [[sc 0 0];[0 sc 0];[0 0 1]] * [R_2(:,1:2),T_2];
```

Generate a set of random feature points and map these into position 1 and position 2 images.

```
px = 3*randn(K,1);
```

```
py = 3*randn(K,1);
```

```
% Determine the images as seen in camera 1 and camera 2 positions
x1 = zeros(K,1);x2 = zeros(K,1);y1 = zeros(K,1);y2 = zeros(K,1);
for k=1:K
    g1 = H_1*[px(k);py(k);1];
    x1(k)  = sc*g1(1)/g1(3);
    y1(k)  = sc*g1(2)/g1(3);
    g1 = H_2*[px(k);py(k);1];
    x2(k)  = sc*g1(1)/g1(3);
    y2(k)  = sc*g1(2)/g1(3);
end
```

Generate the $a^j = x_1^j \otimes x_2^j$ components and put them into the matrix $A = \begin{bmatrix} a^1 & \cdots & a^K \end{bmatrix}^T$

```
A = zeros(3*K,9);
for k=1:K
    a=kron([x1(k);y1(k);1],[[0 -1 y2(k)];[1 0 -x2(k)];[-y2(k) x2(k) 0]]);
    A(((k-1)*3+1):k*3,:)=a';
end
```

Solve the SVD of

$$AH^s = 0 \tag{39}$$

as

```
[V,L,U] = svd(A);
H = zeros(3);
H(:) = U(:,9);
g1 = svd(H);
H = sign(H(3,3))*H/g1(2)
```

Extraction of R_r and $\frac{1}{d}T_r$

Use fsolve() to determine R_r and $\frac{1}{d}T_r$ from H given $\{d, N\}$.

```matlab
options=optimset('Display','iter');   % Option to display output
% initial guess set to first location
x = [0;0;0;Tr;1];
[x,fval,exitflag,output,jacobian] =
fsolve(@(xx)Hr2RrTrfun(H,xx,d,[0;0;1]),x,options);
D=x;
a_est = x(1:3)
Tr_est = R_2'*x(4:6)


function F=Hr2RrTrfun(Hr,x,d,N)
ax = x(1);ay = x(2);az = x(3);
Tr = x(4:6);s = x(7);
e = Hr-Hrmat(ax,ay,az,Tr,s,d,N);
F=zeros(9,1);
F(:) = e;
function H = Hrmat(ax,ay,az,Tr,s,d,N)
 % d  is the perpendicular distance from ZW=0 plane to position 1
 % N is the normal vector of the plane ZW=0 in pos 1 frame
% generate the RT matrix
Cx = cos(ax);Sx = sin(ax);
Cy = cos(ay);Sy = sin(ay);
Cz = cos(az);Sz = sin(az);


Rx = [[1,0,0];[0,Cx,Sx];[0,-Sx,Cx]];
Ry = [[Cy 0 Sy];[0 1 0];[-Sy 0 Cy]];
Rz = [[Cz Sz 0];[-Sx Cz 0];[0 0 1]];
Rr = Rz*Ry*Rx;
H = s*(Rr+Tr*N'/d);
```

3.8 Experimental results of 3D ego-motion based on Tracking fiducial markers

In this section some of the experimental results will be given that are based on the tracking of fiducial markers. The concept is illustrated in a smaller scale in the photo. The handheld naviagation device (HND) observes some fiducial markers in its FOV. It can generate the perspective matrix of the relative 3D position of the camera with respect to the fiducial marker.
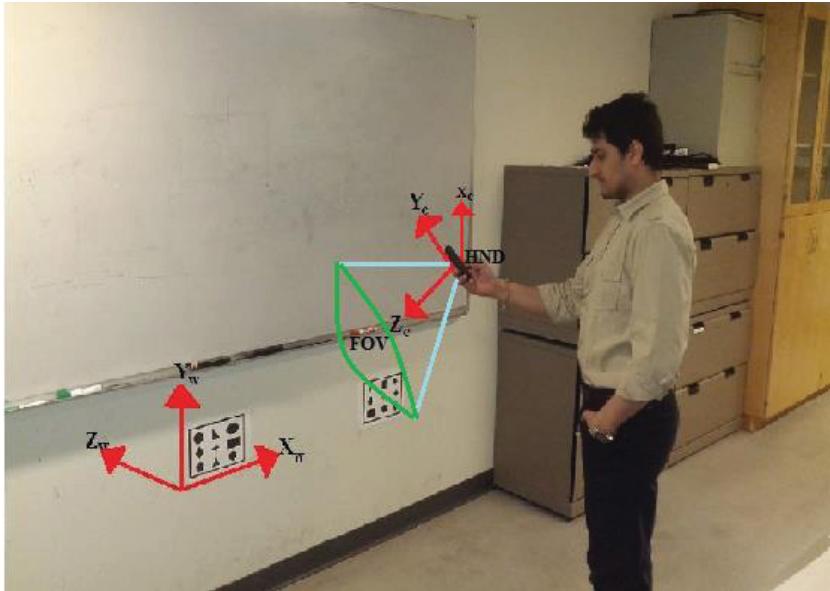
Figure 3.12 *Figure depicting the camera and world coordinate system*

As the camera is moved, the initial fiducial will leave the FOV but other fp's on the same surface will be observed as illustrated in Figure 3.13. The perspective transformation determined based on the first fiducial can be used to locate the fp's of opportunity seen in the FOV at the same time. The fp's located are then used as an equivalent fiducial marker and so forth. Eventually another calibrated fiducial marker known to the camera will emerge in the FOV and the drift can be corrected.

FM   Figure of opportunity   FM

HND trajectory
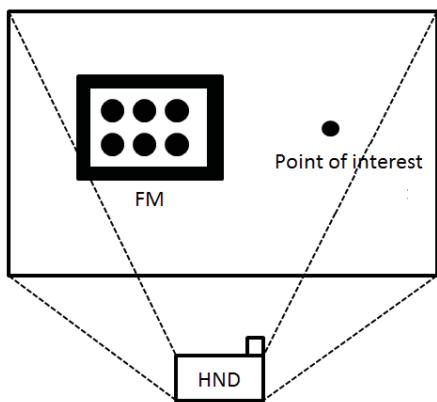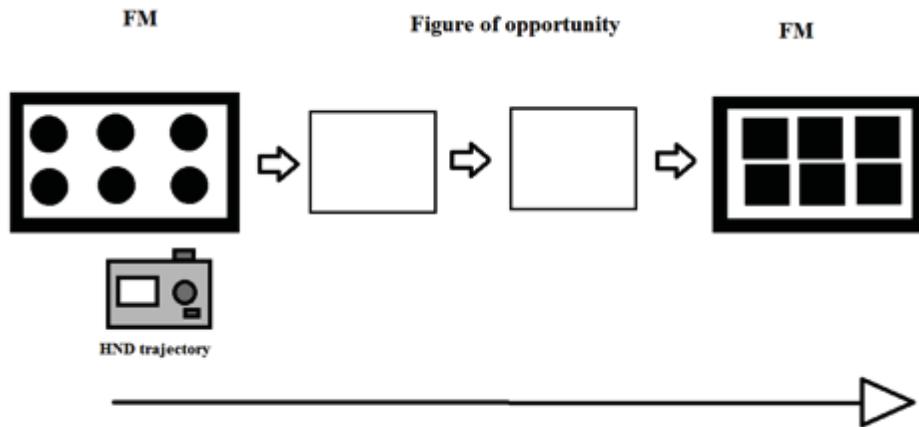
Figure 3.13 *Camera is translating illustrating a continual selection of fp's in the FOV*

Figure 14 shows an image of a fiducial marker that is seen in perspective and corrected after the transformation has been determined and applied. The projective transformation that is determined is then used to infer the ego-motion of the camera itself relative to the fiducial marker.
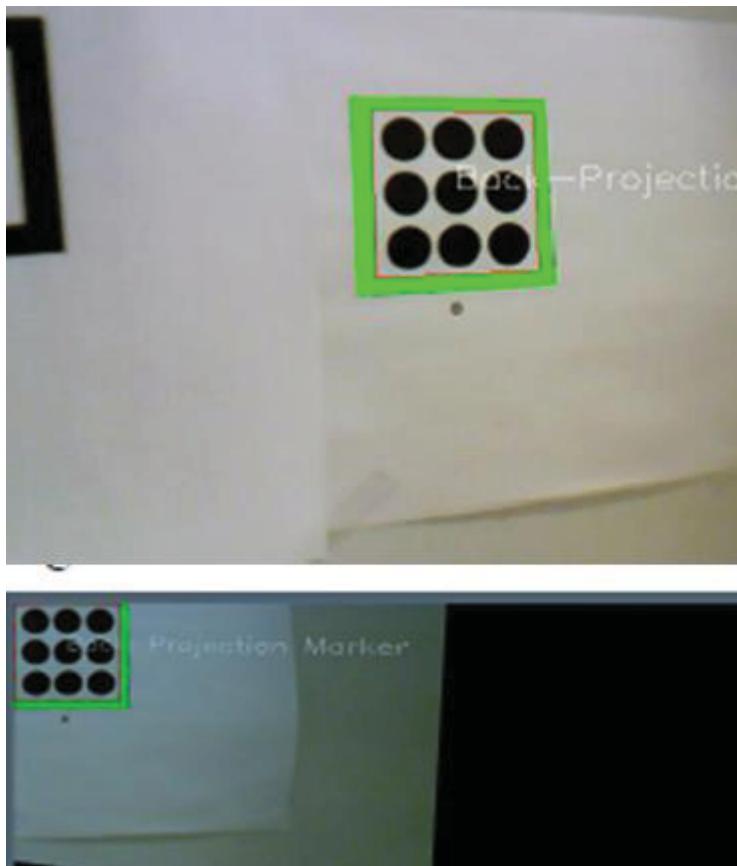
Figure 3.14 *Top image is fiducial in perspective from which the projective transformation is determined and applied resulting in the transformed orthographic view*

The setup for the trial with the sequence of fiducial markers and posed fp's of opportunity is shown in the photo of Figure 3.15.
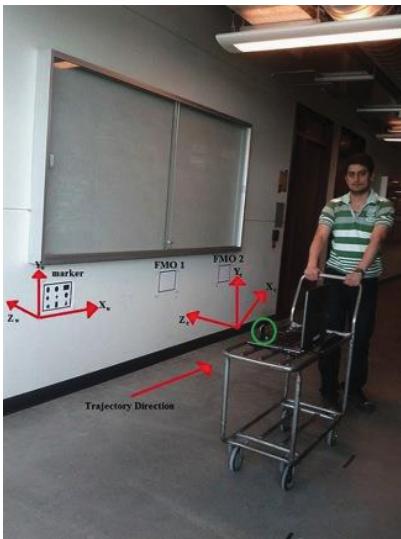
Figure 3.15 *Experimental setup for the sequence of fiducial markers and fp's of opportunity*

The results of this trial is shown in Figure 3.16 for one of the runs. The results show some drift that occurs as well as the issue with the small discontinuities in the solution as the fiducial markers enter and leave the FOV. The drift is an annoyance but it is not large and hence is manageable. When other markers are added, the drift will correct itself. Also the CV method is not intended as the only source of navigation measurements. Wireless sources, as explained in the previous chapters are less accurate over the short trajectory but do not suffer from the same drift issue.
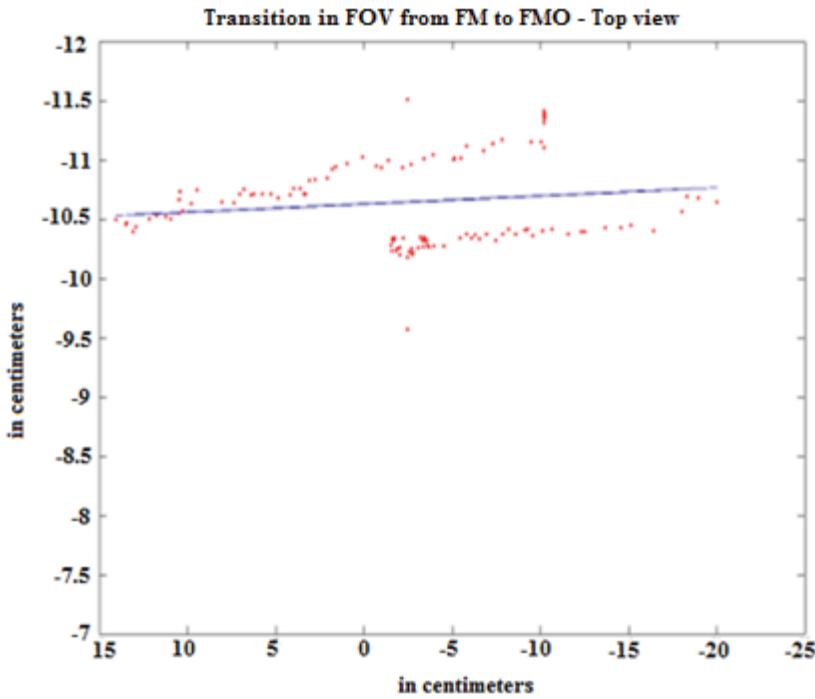
Figure 3.16 *Experimental outcome*

Outcome of a longer length trial is shown in Figure 3.17 in the same hallway setting. Drift is still present but the positioning from the 3D projective is useably accurate.
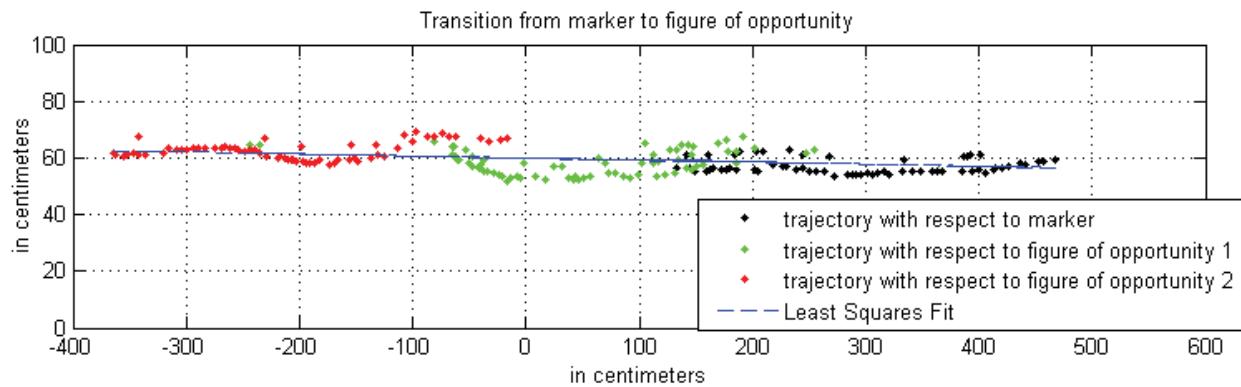
Figure 3.17 *Experimental outcome*

A least squares fit is applied to the trial results of Fig.3-17 and are given in Figure 3-18. The least squares is fit to each segment with the same set of fiducial markers and features. The overall least squares shows very little drift. As the camera was mounted on a cart of constant

height from the floor surface, the height should ideally be constant which it is approximately with very little drift with respect to this variable.
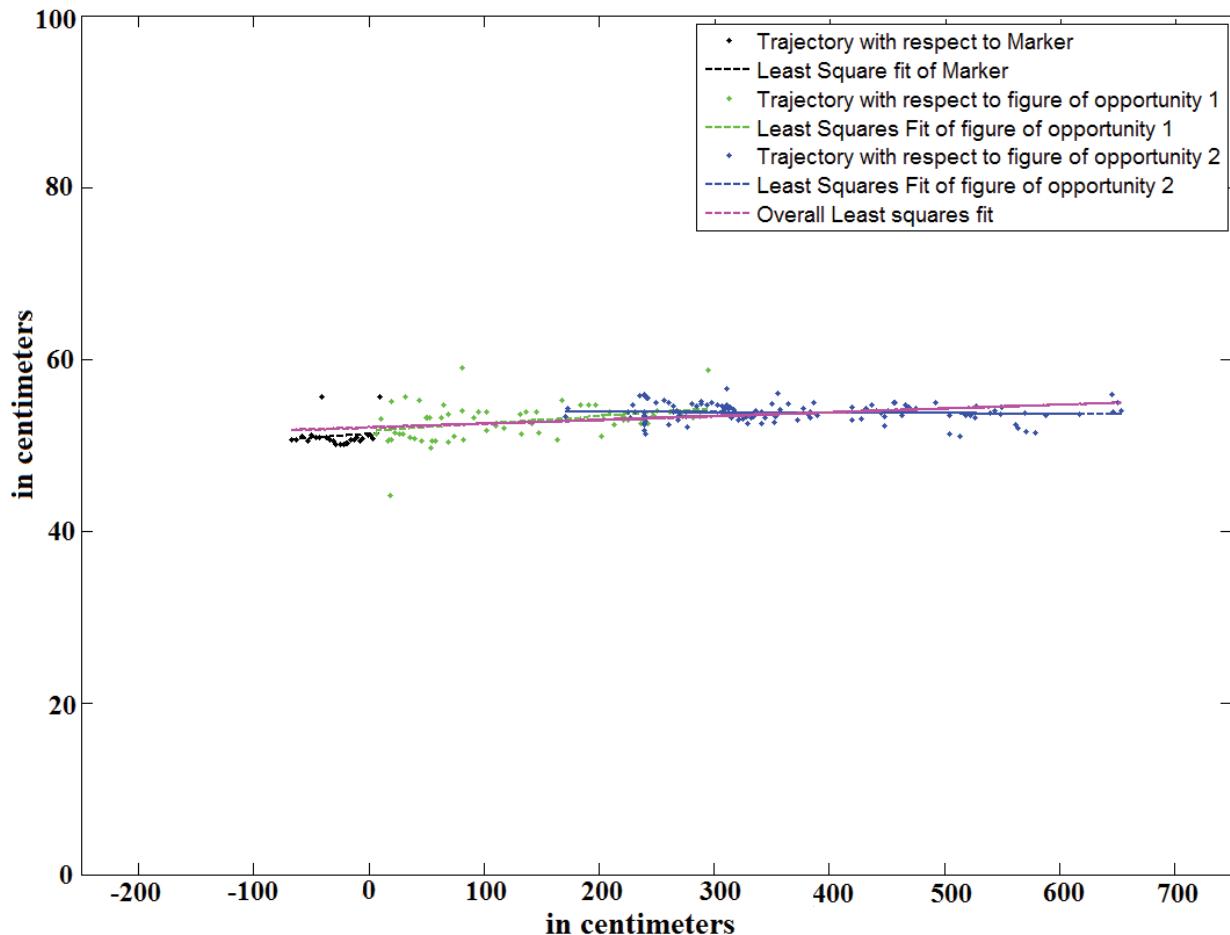
Figure 3.17 *Experimental outcome after least squares line fit*

Having the camera determine the various projective transformations the observations after each frame will be entered into the EKF based on a linear model for the camera trajectory as outlined in the previous chapter.   The processing block diagram is given in Figure 3.18.
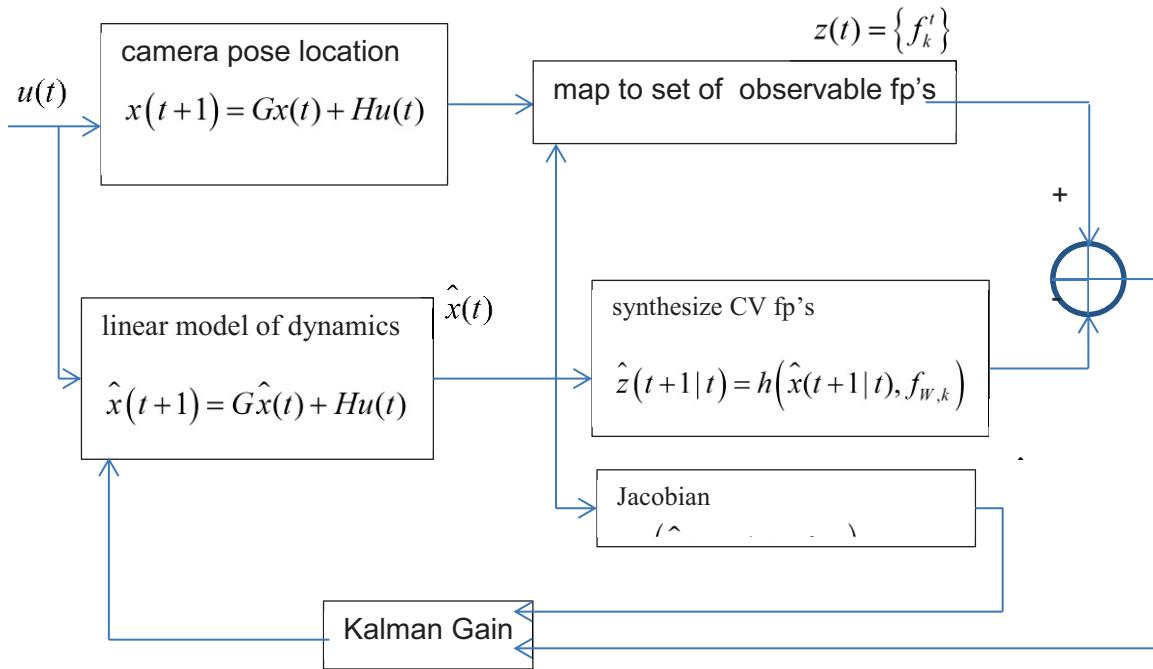
Figure 3-18 *Kalman filter processing used to determine the least squares trajectory of the HND*

3.9 Extracting Lines with Hough Transforms

In addition to the fp's of the wall or floor surface and fiducial markers it is convenient to use rectangular structures such as windows, doors and picture frames as well as lines that exist in every room. As projective transformations of lines in 3D are lines in the image plane such structures are preserved. Figure 3-19 illustrates some of the features and lines that are available in a 3D scenario.
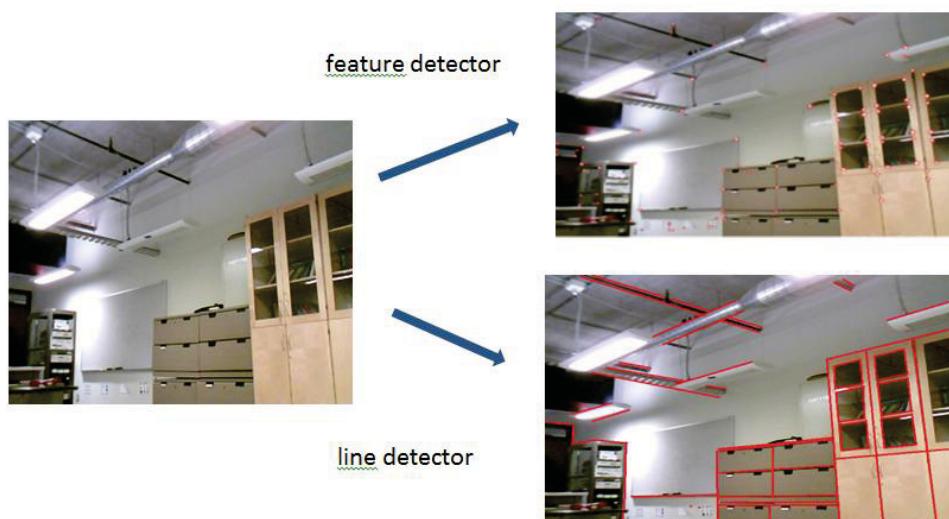
Figure 3-19 *Typical indoor setting has extensive instances of lines and feature points.*

Geometric structures such as lines, rectangles and circles can be found in a camera image using the general Hough transform It is essentially a transform that determines the likelihood of the geometric construct appearing in the image over the range of the unknown variables. For instance, consider a line which is defined by two parameters, the slope angle and skew distance from the origin. The Hough transform would map the image in x-y to the two dimensional uknown parameter space of slope angle θ and skew distance ρ . A transform point of (ρ,θ) then is an indication of the likelihood that a line with the parameters of (ρ,θ) exists in the image.   For a Hough ellipse transform, there are five parameters. Usually the transform is a construction of two of these parameters such that the other three are given. For instance the two parameters could be the center of the ellipse with the major and minor axis given as well as the orientation angle. This is better shown with worked examples of the line and the circle.

As shown in the figure below, any line segment that is extended into an infinite line can be described by two variables:

ρ - skew distance from the origin to the line

θ - angle of the skew line

For each point in the intensity field plot a contour of all the potential $\{\rho,\theta\}$ combinations or lines that the intensity point could belong to. This line is weighted by the intensity of the point. In this way when sweeping over an intensity field, the points corresponding to a bright line of parameters $\{\rho,\theta\}$ will add up to form a peak in the 2D intensity field of $\{\rho,\theta\}$. This intensity plot is then thresholded to yield the estimate of the $\{\rho,\theta\}$ of the line segment. Note that the longer and more intense this line is, the higher the peak will be.
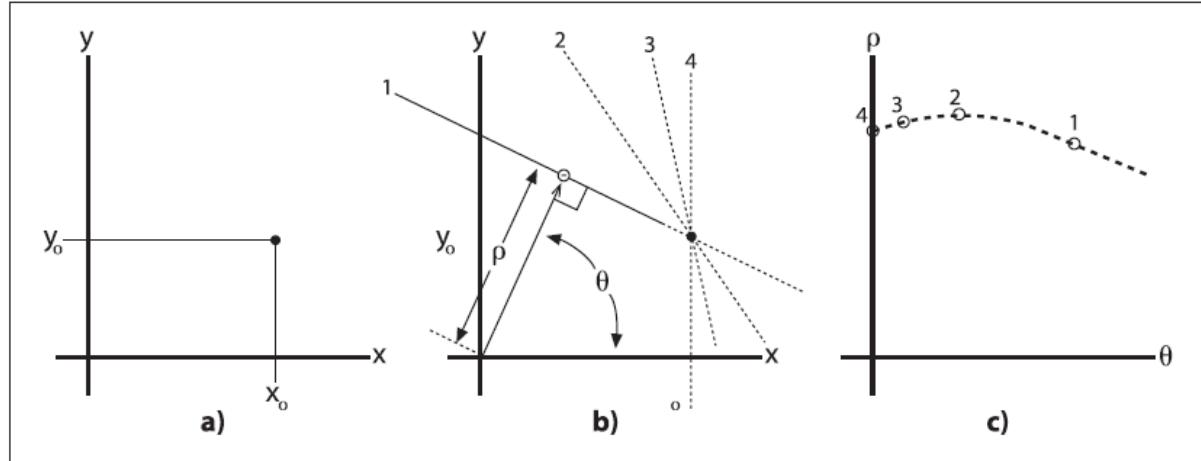
Figure 6-10. A point (x_0, y_0) in the image plane (panel a) implies many lines each parameterized by a different ρ and θ (panel b); these lines each imply points in the (ρ, θ) plane, which taken together form a curve of characteristic shape (panel c)

Figure 3-20 *Hough line detection (taken from Learning OpenCV Bradlie)*

An example of using the Hough transform for lines based on the image in Figure 3-21 is given. First the image is converted to grey scale and then the Canny edge detector is applied (taken from Matlab, Image processing toolbox user guide.

Figure 3-21 *Image of gantry section*
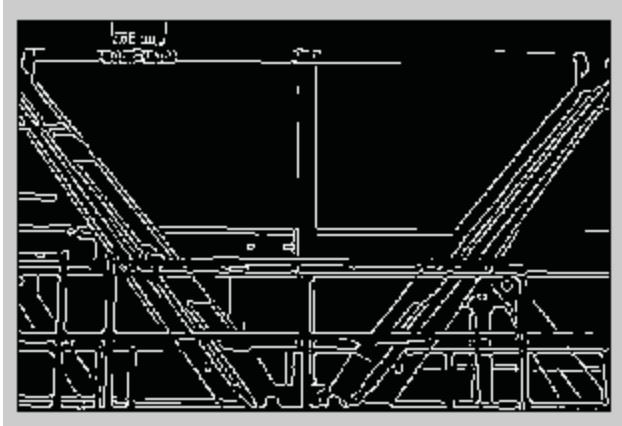
Figure 3-22 *Edge detection of gantry section*

Next take the Hough line transform of the edge image which is shown in Figure 15.

```
[H,T,R] = hough(BW);
figure(4);imshow(H,[],'XData',T,'YData',R,...
           'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal;
```

H is the likelihood as a function of ρ and θ. It can be plotted as a mesh or contour which sometimes provides greater clarity. The highest peaks of H which correspond to the most probable lines are given by houghpeaks which can be added to the image using

```
hold on;
P  = houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))));
x = T(P(:,2)); y = R(P(:,1));
plot(x,y,'s','color','white');hold off;
```
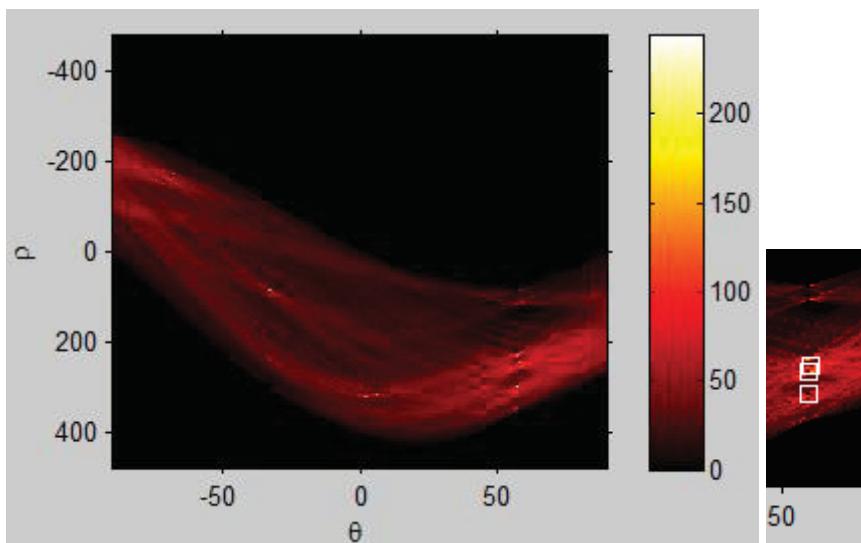
Figure 3-23 *Hough transform of edge image with highlighted peaks around* $\theta = 50^o$

Now find the lines and map them onto the figure

```
% Find lines and plot them
lines = houghlines(Ic,T,R,P,'FillGap',5,'MinLength',7);
figure, imshow(I), hold on
max_len = 0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

   % Plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

   % Determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end
end

% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','blue');
```


Figure 3-24 *Hough lines mapped back into image with starting and stopping positions*

Note that there are five lines but that each line consists of multiple segments. Hence hough lines finds the line segments within the line. The segmented line is an additional step beyond the Hough transform. Line segments are based on the intensity of the edge detection of the image itself.

Figure 3-25 *Hough Lines applied to the building image*

An interesting development for indoor navigation is based on tracking floor tiles using Hough transform and projective transformations. Initially try the features of opportunity in the video which is not successful as seen in Figure 3-26.
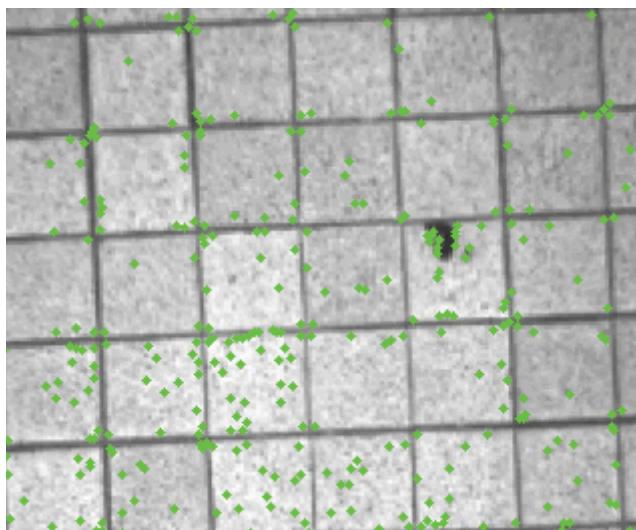
Figure 3-26 *Tiled floor surface with corners() applied to determine features of opportunity*

There is an obvious difficulty in establishing the movement of features. After some careful manual adjustment of the threshold gives us some hope of using a better set of feature points that can be used for differential movement as shown in Figure 3-27 where the green and red

fp's indicated are for two consecutive image frames. Clearly it is not a robust solution, especially as manual search for the optimum threshold is required.
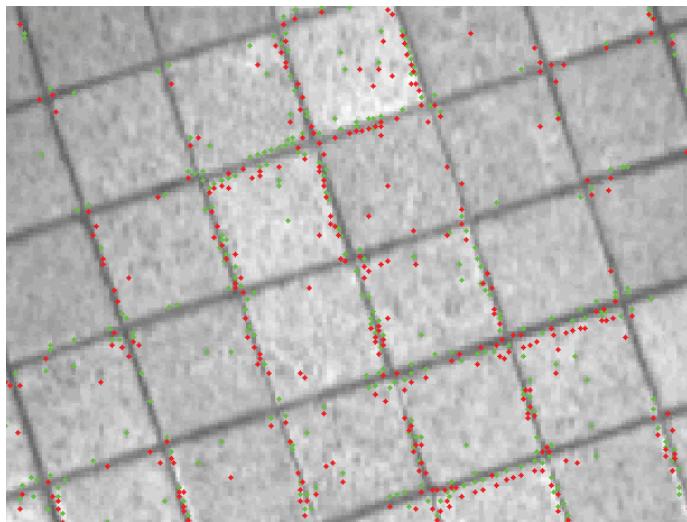
Figure 3-27 *Tiled floor with fp's of two consecutive frames*

With Gaussian spatial filtering and tweaking of quality threshold we can get fp's from the corners of the grout lines. A problem is that the fp's jump around as the metric for the fp's has a saddle point in each grout line intersection making the location of the fp very sensitive to any noise or neighbourhood clutter.
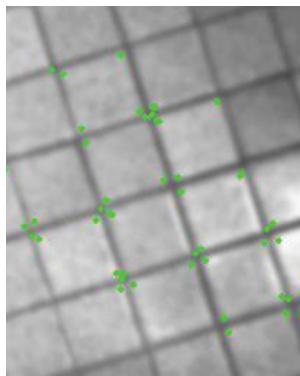
Figure 3-28 *Tiled floor feature points after Gaussian filtering applied*

Next we try the Hough line transform and we get the plot of the transform in Figure 3-29 and an overlay spatial image as in Figure 3-30.
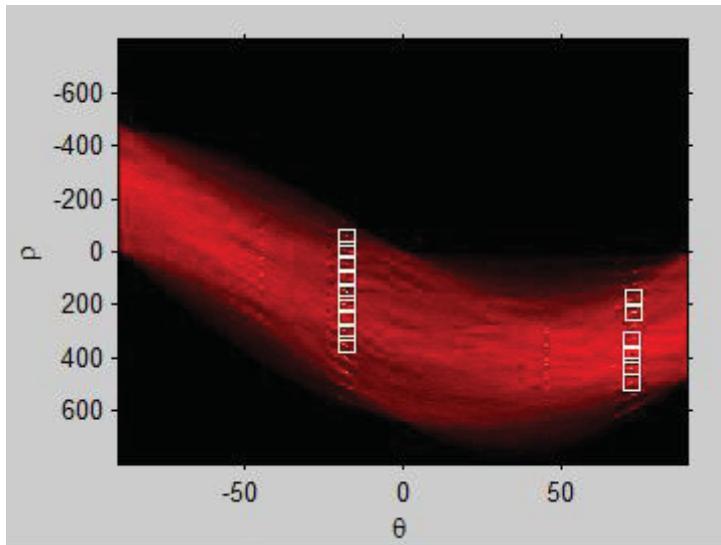
Figure 3-29 *Hough line transform output for tiled floor image showing sets of peaks indicating the vertical and horizontal lines*
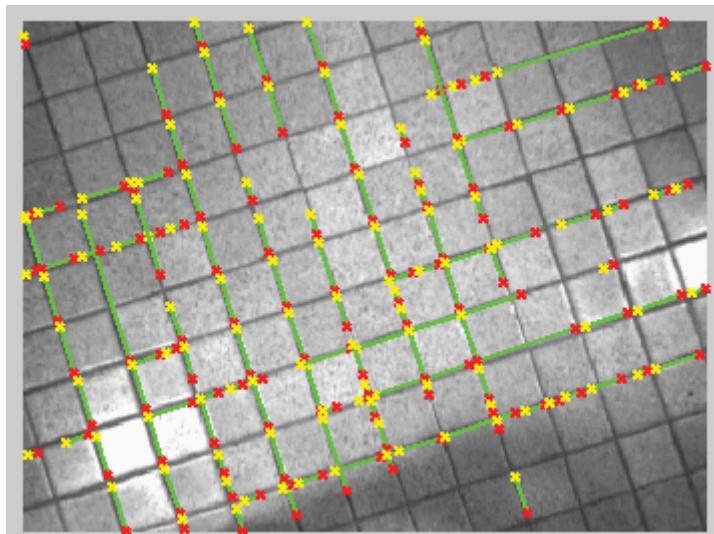
Figure 3-30 *Hough line transform output for tiled floor image*

Next we want to extract the infinite lines. This works well if we can assume that the camera is well calibrated and that the tiling is accurate. Note that the Hough line output is P which are indices of T and R. and T are

```
figure(4), imshow(tile_vid(:,:,1)), hold on
NL = length(P);
Nx = 640;Ny = 480;
```

```matlab
for j=1:NL
theta = T(P(j,2))*pi/180; roe = R(P(j,1));
xo = roe*cos(theta);yo = roe*sin(theta);x1 = -sin(theta);y1 = cos(theta);
pp = zeros(4,3);
% Solve for x=1 intercept
t = (1-xo)/x1;
ya = yo+t*y1;
if ya>0 & ya < Ny
    pp(1,:) = [1,ya,1];
else
    pp(1,:) = [1,ya,0];
end

% Solve for x=Nx intercept
t = (Nx-xo)/x1;
yb = yo+t*y1;
if yb>0 & yb < Ny
    pp(2,:) = [Nx,yb,1];
else
    pp(2,:) = [Nx,yb,0];
end

% Solve for y=1 intercept
t = (1-yo)/y1;
xa = xo+t*x1;
if xa>0 & xa < Nx
    pp(3,:) = [xa,1,1];
else
    pp(3,:) = [xa,1,0];
end

% Solve for y=Ny intercept
t = (Ny-yo)/y1;
xb = xo+t*x1;
if xb>0 & xb < Nx
    pp(4,:) = [xb,Ny,1];
else
    pp(4,:) = [xb,Ny,0];
end


% Plot the line
ppp = [];
for jj = 1:4
    if pp(jj,3) == 1;
        ppp = [ppp;pp(jj,1:2)];
    end
end

plot(ppp(:,1),ppp(:,2),'LineWidth',2,'Color','green');
end
hold off;
```
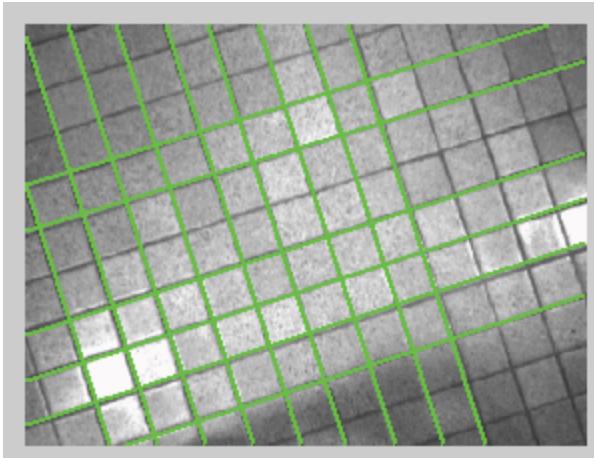
Figure 3-31 *Hough line transform output for tiled floor image showing grid of infinite lines*

With the Hough lines the intersection points can be calculated for the grid of tiles. Note that each Hough line is an average of all of the points that have a higher likelihood of belonging to the line. Hence the method is much more robust than searching for fp's of the grout line intersections.

The robustness of the hough line method is observed in the following where a rectangular paper is imaged with it's corners removed. Hough line transform has no issue with determining the virtual corners of the rectangle that would otherwise have been present.
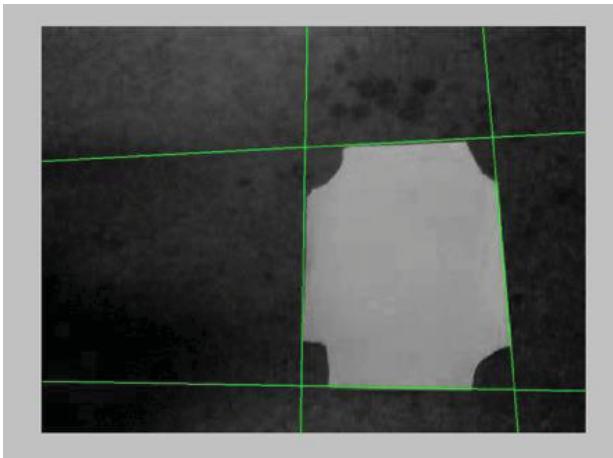
Figure 3-32 *Hough line transform and infinite line intersection applied to a rectangle with corners missing.*

4. SGL Experimental Apparatus

4.1 Overview

As described in the previous chapter, simulation of the SGL with the SA is limited due to the difficultly of justifying the propagation model in a multipath environment. Essentially the performance results of the SA are readily predictable given a specific multipath model. However, to access the actual performance of the implemented SA based SGL it is necessary to have a set of statistical measurements.

Measurements based on spread spectrum modulated signals will be made with chipping rates of up to 20 Mcps. From these the channel impulse response (CIR) can be measured. Having up to four transmitter antennas in various locations, the simultaneous multipath propagation channels can be represented which are difficult to model justifiably. The other main objective of the measurement system is that of the LOS detection and LOS processing which get's to the main contribution of the SA.

A wireless system consisting of several transmit antennas and a receiver that can be moved while taking measurements has been created and is used for the experimental component of this contract. This wireless system can emulate the transmission, reception and processing of the LTE downlink signal reference signals that the SGL. The carrier frequency can be tuned within the operating range of 700 MHz to 2.2 GHz with various modulation options from a simple tone to a full 20 MHz bandwidth modulation. In phase III the objective will be to expand this bandwidth further to 100 MHz that will account for the evolution of the LTE standard. Modifications can be made to the existing hardware system to provide this. This will involve the receiver and transmitter using a step frequency such that the 100 MHz is covered by multiple bands each of 20 MHz instantaneous bandwidth. Figure 5.1 shows a typical configuration in the experimental room of ENA401 at the University of Calgary. The idea is to locate the 4 antennas such that they have a variable amount of multipath and in some cases, as the trajectory is changed that the propagation will vary from LOS to NLOS.
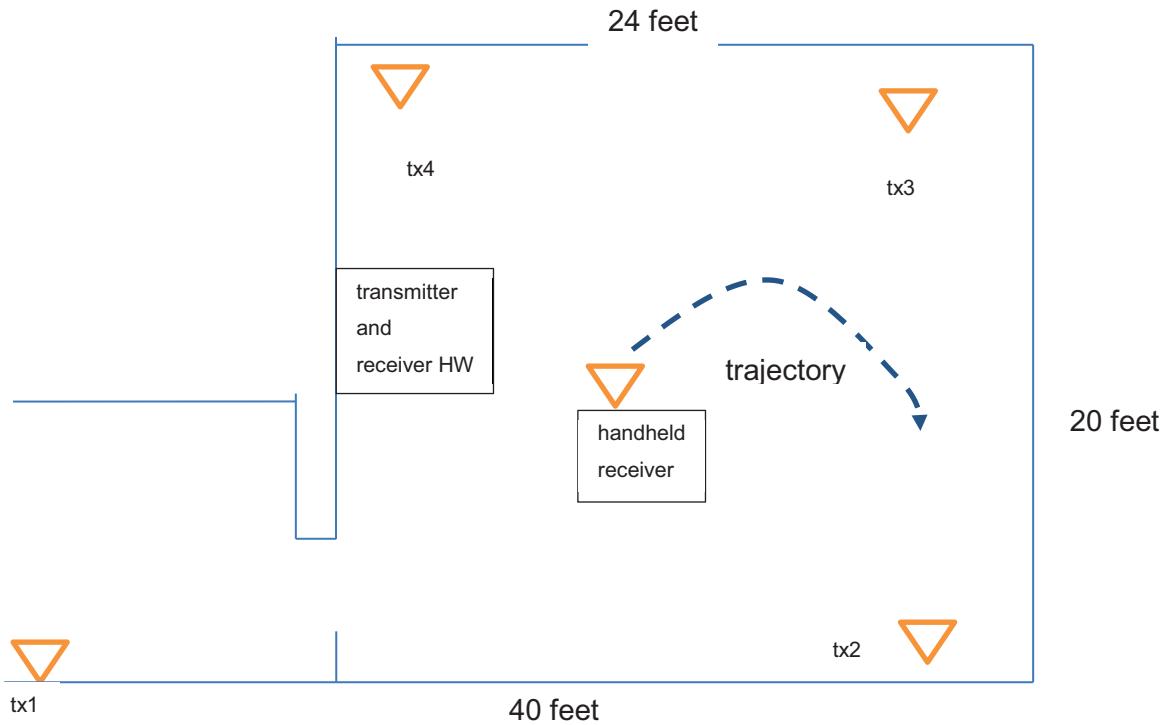
Figure 4-1 *Floor plan layout of the LTE propagation room ENA401 at the University of Calgary*

The wireless measurement system that was developed will be described in section 4.2. Section 4.3 describes the processing of the single receiver channel with a reference antenna channel. Section 4.4 describes the multi-antenna setup and the overall test environment.   Section 4.5 provides experimental results and Section 4.6 an overall summary and some conclusions.

4.2 Wireless measurement system

The wireless measurement system developed essentially consists of a set of four AN's and a UE. Each AN consists of an RF amplifier and antenna that is mounted to the wall in the test area. The UE consists of a small antenna and LNA for receiving the AN transmitted signal. It is also bundled with a webcam for the associated CV measurements.   The transmit signal is generated by a National Instrument software radio transceiver, denoted as the USRP. A similar matching USRP is used in the receiver to facilitate the processing.   A block diagram of the overall architecture of the system is given in Figure 4.2.

up to 4 tx antennas

Tx switch

USRP TX

USRP RX

PC config operate

turn table

PC operate Rx

10 MHz source

webcam

laptop

trigger
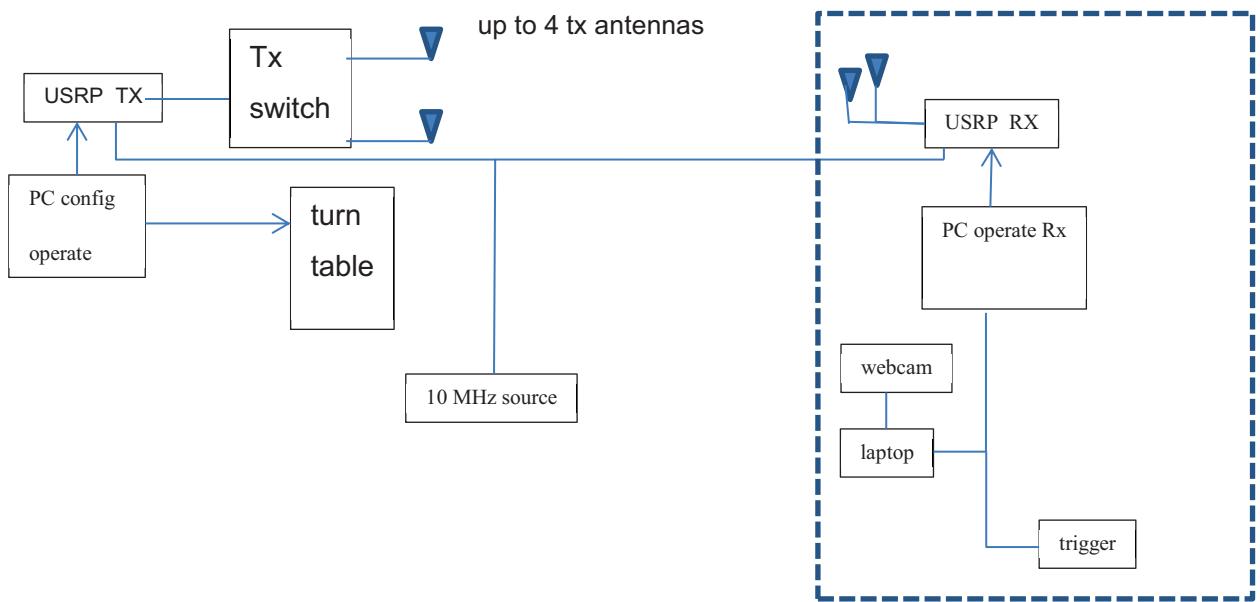
Figure 4.2 *Overall block diagram of the LTESA hardware system*

The USRP's for the transmitter and receiver functions are programmed via LabView. This provides maximum flexibility for the various experimental setups that must be implemented. As LabView is a rather cumbersome software development environment with discovered limitations, it will only be used for the signal modulation for the transmitter and the sample extraction in the receiver. The SA processing as well as the general SGL processing will be done off line via Matlab and C. In Figure 4.1 it is observed that the receiver has two antennas. One is used as an optional reference antenna and the other antenna is main SA receiver antenna. The receiver has a separate laptop that is used to run labview to collect the data from the USRP receiver and to collect the CV data. The transmitter is run from a separate laptop and USRP. The reason for this is that the receive and the transmitter can be physically separated without the inconvenience of having to tie them together with data connections. Also there is a 10 MHz synchronizing signal that is used to frequency lock the receiver and the transmitter. This is so that the frequency instability of the receiver can be removed in some experiments to study impacts. An additional laptop may be required for the CV processing of the webcam output. The issue is that if a single PC services boththe webcam and the USRP then the connections are a bottleneck resulting in issues of irregular sampling such that the camera and wireless data are not easily synchronized. Additional components for the system are the

precision turntable which is used to move the antenna as an alternative to hand motion. The precision turntable motion implies that the webcam is not needed as the antenna trajectory is easily calibrated. This turntable is driven by the PC used for the receiver. Additionally, the receiver PC controls the selection of up to four transmit antennas as shown in Figure 4.3. Shown is the Digital Input Output (DIO) interface block connected to the receiver PC via a USB that provides the output digital signals for sequencing the antenna, turntable stepper motor and receives inputs from the trigger switch.
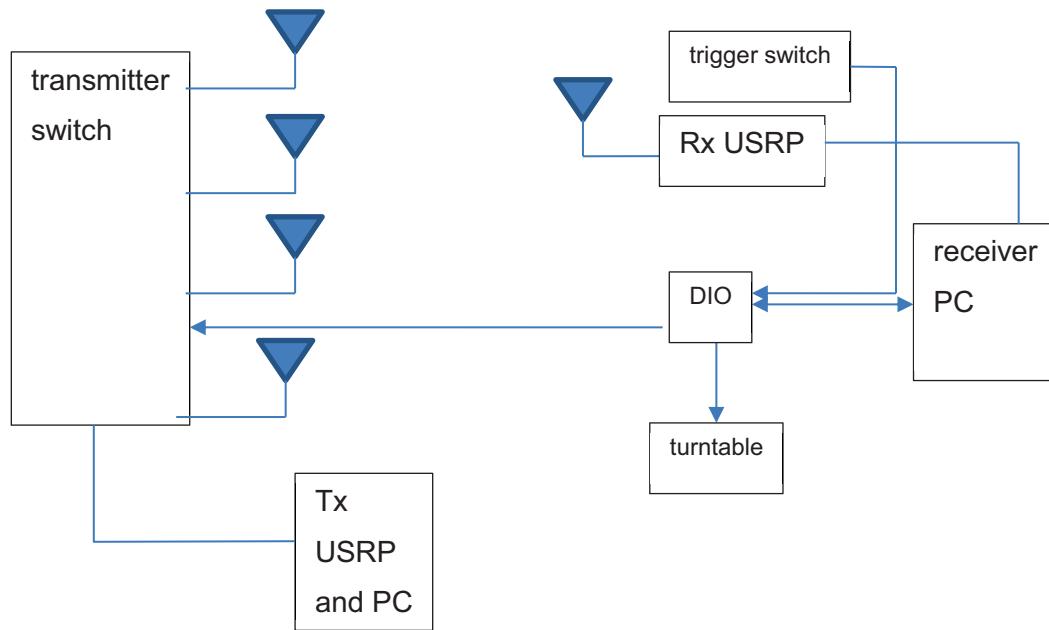
Figure 4-3 *Block diagram of the antenna selection, trigger and turntable*

A block diagram of the USRP is given in Figure 4.4. It consists of a transmitter and a receiver channel that can be switched to the antenna ports. In the transmitter, generic quadrature modulation is used with a complex sampling rate of up to 33 MHz. Likewise in the receiver quadrature demodulation is used with a sampling rate of up to 33 MHz and 16 bits of quantization. A gigabit Ethernet connection is used for all of the data exchange between the host PC and the USRP. This is a limitation as the full modulation bandwidth at 16 bits I and Q in addition to the Ethernet protocol overhead exceeds the 1 Gbps capacity of the connection. Hence if higher IQ sampling rates are used then it is necessary to limit the sampling block size. The USRP has internal buffering such that high rate burst sampling can be accommodated provided that the average data traffic required is within the 1 Gbps limitation. The limitations of

the Ethernet connectivity is also the reason why two PC's are used, one dedicated for the transmitter and the other for the receiver.
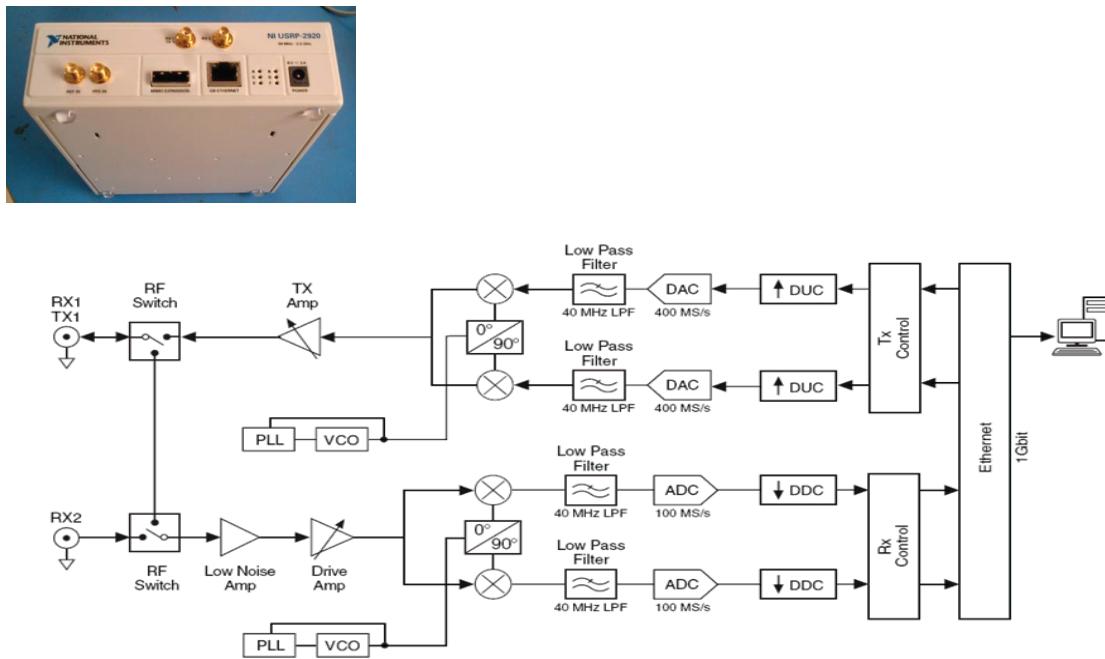
Figure 4.4 Block diagram of the USRP

The Labview programming of the receiver USRP is a simple state machine. Unfortunately due to the buffering of the data it is not possible to precisely control measurement segments of the USRP. Rather it samples for the 4.5 seconds at a rate of 200,000 samples per second which are then transferred to the hard-drive of the receiver PC. Hence the turntable and receiver antenna sampling are driven by a separate process. The USRP and turntable/antenna switch are run as two separate processing threads. The states of the LabView program for gathering the USRP data is simply an initialization phase, sampling phase, data storage phase and a termination phase that are run in sequence. An issue with the USRP is that every new measurement interval implies a new initiation with causes the LO to re-lock. This uncontrolled phase cannot be tolerated in the SA.

Prior to starting the sampling, the frequency needs to be adjusted. If the rx is offset by 100ppm then the result is $1575.42MHz \times 10^{-5} = 16kHz$. This offset is reduced by adding an offset to the receiver signal. It is reasonable to assume that this can be hardwired in bringing the uncertainty down to about 1 kHz. The minimum sampling rate is about 200 kHz. Hence there is obviously

no issue with aliasing and the input signal can be low pass filtered to 4 kHz and decimated by a factor of 20 to 10 kHz effective sampling rate prior to storage.

There are various hard wired variables used as listed in the following table:

variable	description
samp_rate	Sampling rate set to the minimum of 200k for a CW signal and up to 33 MHz for a spread spectrum signal
fLO	LO frequency set to 1.57542 GHz for the GPS related sampling but can be anywhere from 700 MHz to 2.2 GHz
channel	RX2 is used primarily.
Number of samples	Can be set arbitrarily high. Typical for the CW SA analysis a million samples are used for a 5 second SA epoch.

A brief description of the LabView programs will be given. The LV panel for the stepper for the turntable as well as the receiver antenna is given in Figure 5.5. Note that it is necessary to run these operations from a separated processing thread so that it does not interfere with the USRP sampling operations. Settable controls are the number of events which are essentially steps of the turntable stepper. Every step is set at 8 msec and after four steps (32 msec) the antenna is switched. The turntable direction can be toggled as clockwise or counter clockwise. Finally the turntable can be disabled such that only antenna toggling is used. Figure 5.6 gives the LV block diagram which consists of a single for loop that is incremented every 8 msec. A formula block is used to give the digital outputs to the DIO for the antenna switch and the stepper sequencing.
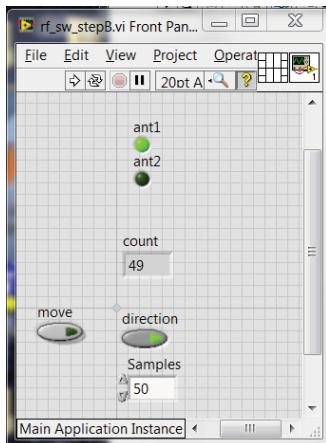
Figure 4.5 Panel of stepper/RF receiver antenna switch control
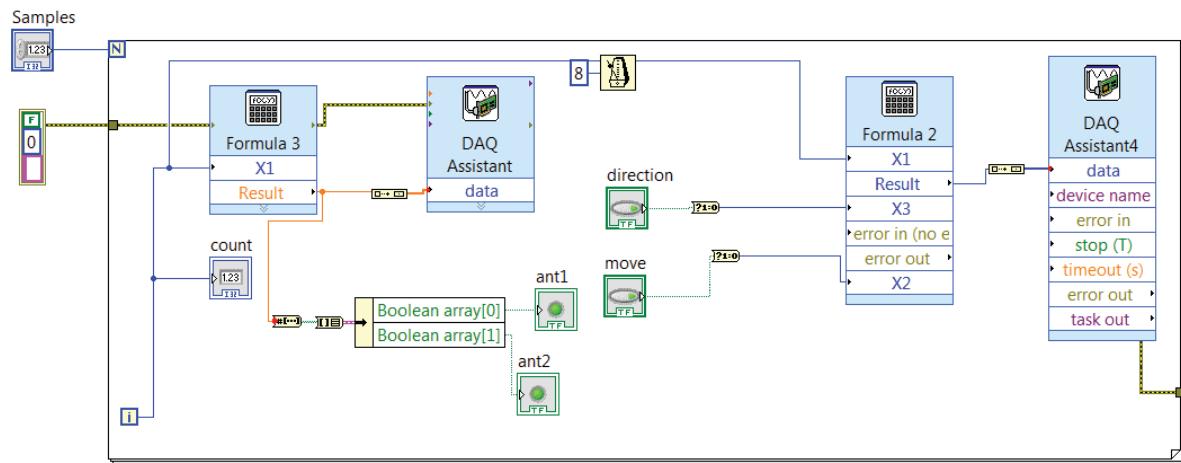
Figure 4.6 LV block diagram of the stepper/antenna switch control

The formula for the stepper sequencing is given by

X2*(mod(X1,4)+4*mod(X1,4)+4+8*X3)

Here X1 is the integer index of the loop, X2 is the boolean variable enabling the turntable to move and X3 is the direction. The four least significant bits are used by the stepper driver. The panel for the USRP control is shown in Figure 5.7. There are no user controls to set. The panel only indicates when the sampling is taking place and the coerced USRP variables. Also shown is the display of the samples which provides an indication of the quality of the sampling. In this instance, the sampling of the reference and the receiver antenna can be seen in the graph. The reference should be the first to start and should be fairly uniform in amplitude. The receiver antenna in this case shows a multipath faded signal as the antenna is moved.
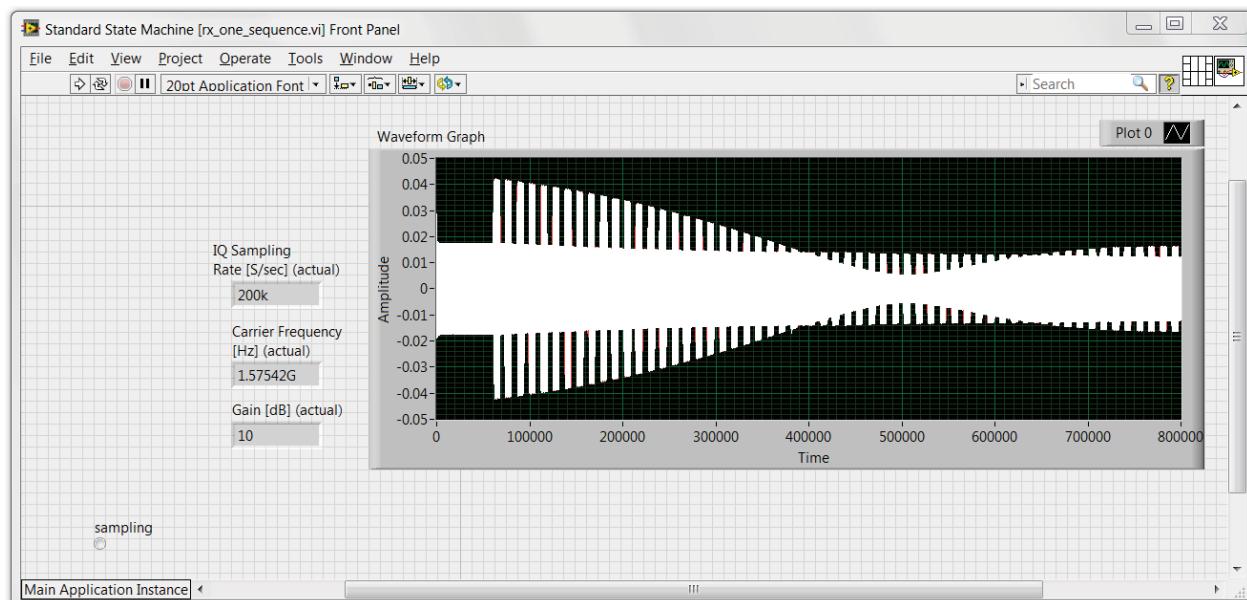
Figure 4.7 USRP Control panel
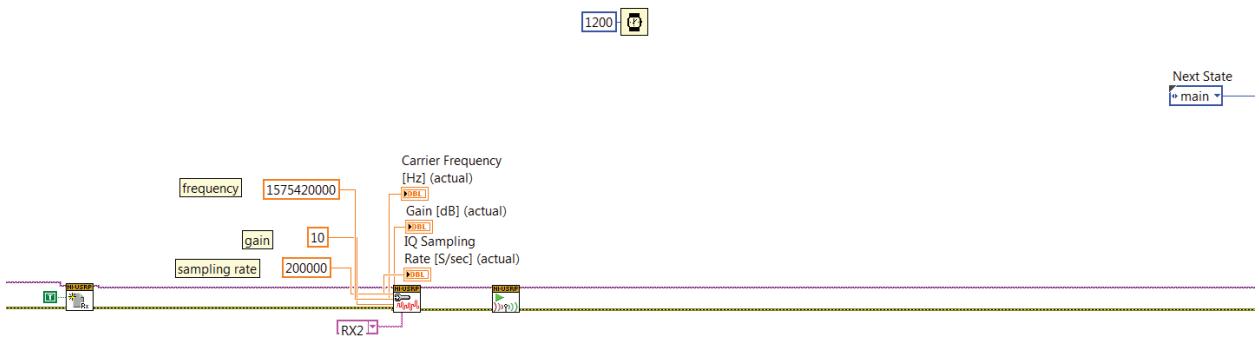
Figure 4.8 USRP Control block diagram for the initialization state.

Figure 4.8 shows part of the initialization of the USRP with the hardwired variables of LO frequency, receiver gain in dB and sampling rate as well as the receiver channel RX2. The actual sampling occurs in this state. Figure 4.9 shows the state for writing the I and Q waveforms to files on the hard drive.
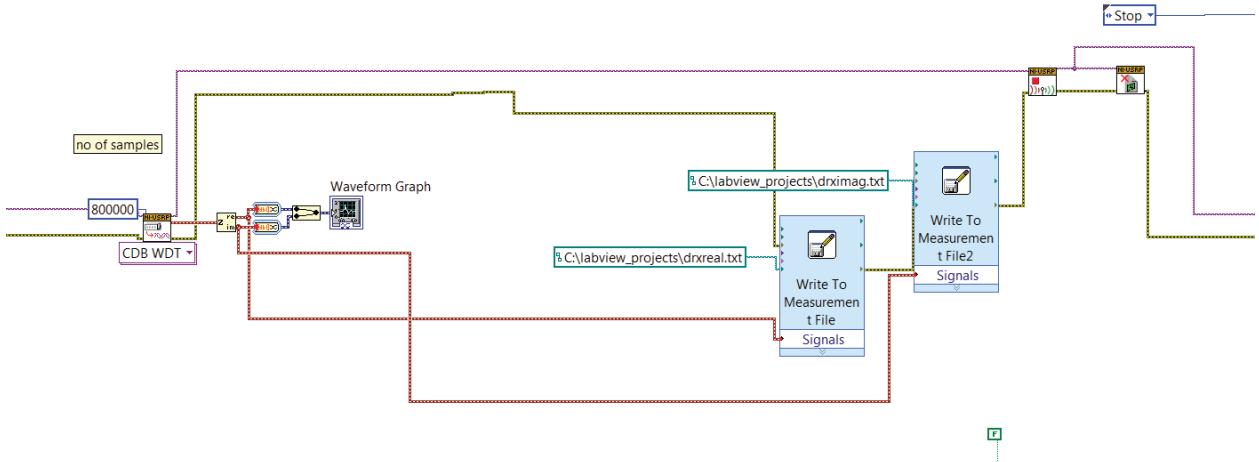
Figure 4.9 USRP Control block diagram for the initialization

Figure 4-10 shows the block diagram of the two receiver antennas, switch and turn table stepper. Figure 4-11 is a picture of the receiver antenna on the turntable.

.
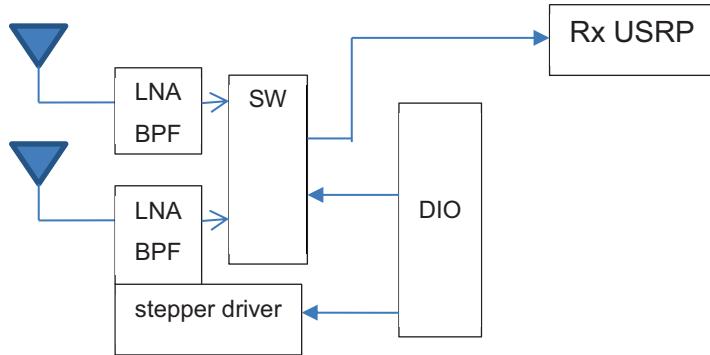
Figure 4-10 *Block diagram of antenna switch and stepper*

rx antenna

reference antenna

web camera pointed to floor

turntable   stepper driver
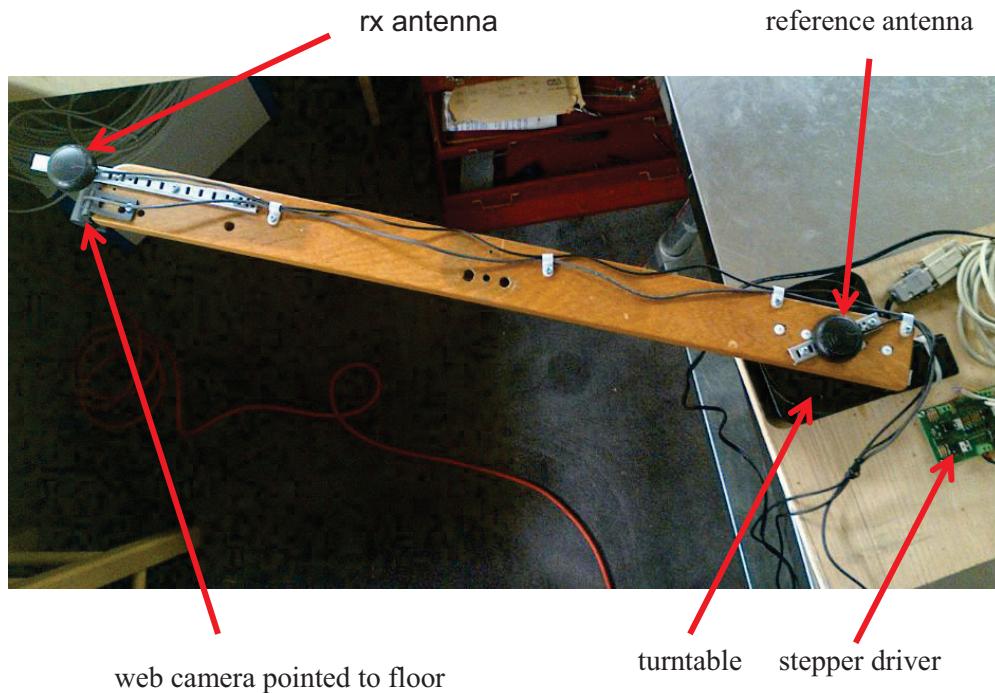
Figure 4-11 *Photo of receiver antenna and reference antenna setup on turntable*

Figure 4-12 shows a close-up of the receiver antenna and the web-cam. The antenna is magnetically mounted so that it can be moved such that it's phase center can coincide with the webcam image center.

Figure 4-12 Close-up of the receiver antenna and the web-cam

4.3 Processing of received wireless signal

As discussed in the previous section the USRP samples the I and Q signals of the antenna output as a contiguous block. The reference antenna and the receiver antenna outputs are commutated by a switch into the USRP. The file of typically 800k samples will be stored on disk and then read into Matlab for offline processing. This processing will be described in this section. The complete Matlab program is listed at the end of the chapter.

Initially the IQ data is read into the matlab workspace and combined into a set of complex samples r. A plot of $|r|$ is shown in Figure 4-13. The transitions are due to the switching between the two antennas which in the present configuration occurs every 32 msec. The slow variation is due to the antenna being moved by the turntable in an environment where there is multipath.
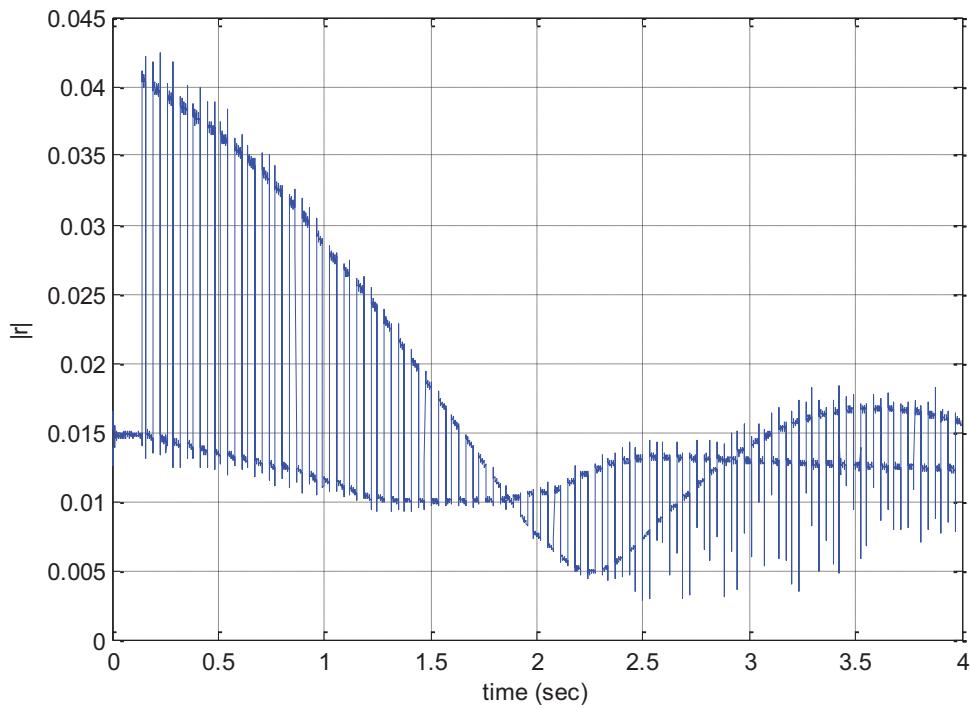
Figure 4-13 Plot of $|r|$ showing the variations of the two multiplexed antenna responses

The first task is to determine the decomposition of the samples of r such that the reference and received channels can be separated. While the timing is not perfect, it is known that the dwell time of the sampling is 32 msec for each antenna. Typically, as in the example shown in 5-13, $|r|$ will be quite different for the two antennas. Hence the processing is to slide a template across with intervals of 32 msec of samples each. The best alignment is when the variance of $|r|$ is the smallest in each interval. Note that if the intervals contain the switch transition then $|r|$ will change abruptly within the interval resulting in a large variance. The average interval variance is denoted by G and is plotted in Figure 4-14 as a function of the offset of the template pattern. There is a well defined minimum which is then used to give the offset required.
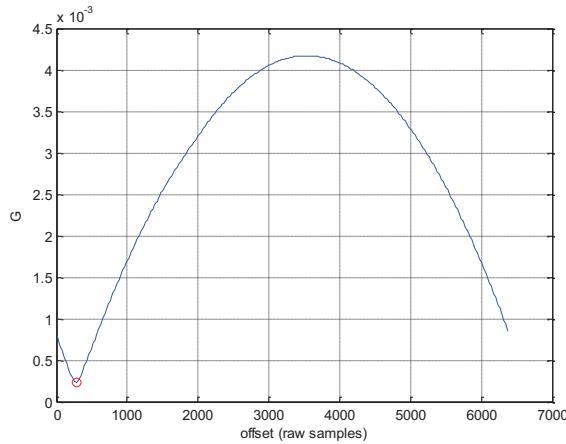
Figure 4-14 Plot of the average interval variance of $|r|$ as a function of the offset

With this offset the data can then be segregated into the two antenna responses. In the current example which is plotted in Figure 4-15 the red samples are for the receiver antenna and the green samples are for the reference antenna.   Note that the green reference samples show less variation as expected.
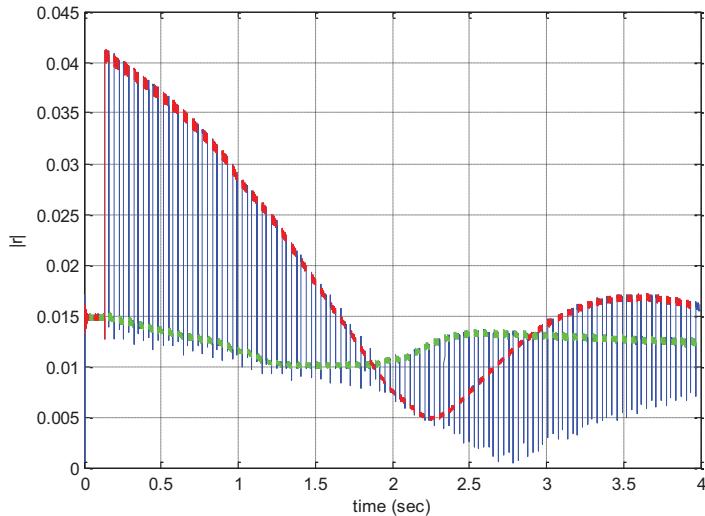
Figure 4-15 Plot of segregated data

As it is undesirable to have to control the timing of the sampling it may not be obvious which is the reference channel. However, this is generally clear from the variations of $|r|$ Also for whatever reason (to be investigated) the first interval is sometimes too short and hence there is a switching transition that occurs within the interval. This results in an interval of samples of high variance that should be discarded. This is shown in Figure 5-16. Figure 5-17 shows a plot of the real and imag components of each sample after they have been derotated in phase

based on an approximate estimate of the offset frequency. Note that only the reference signal samples are used for the estimate of the offset frequency.
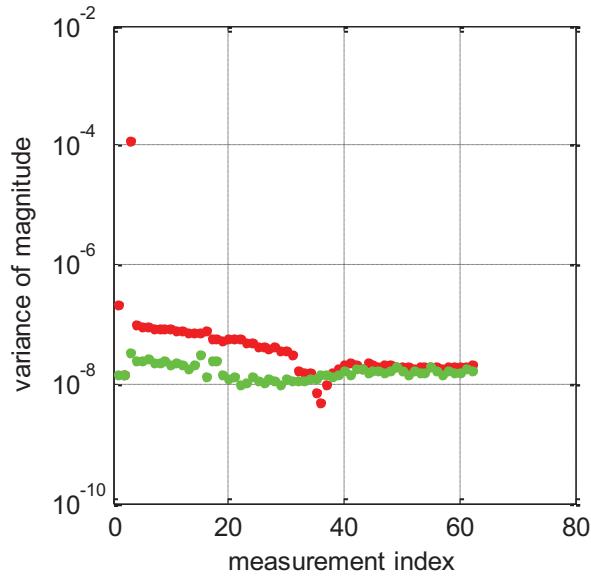
Figure 4-16 Sample of incomplete interval having a high variance of $|r|$
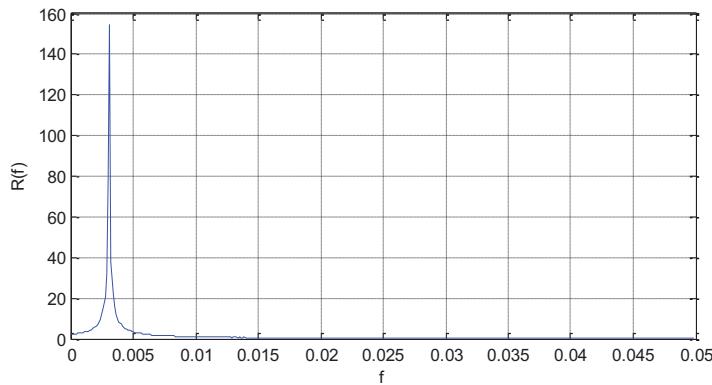
Figure 4-17 Typical example of a FFT of the sample segments indicating the estimation of the offset frequency
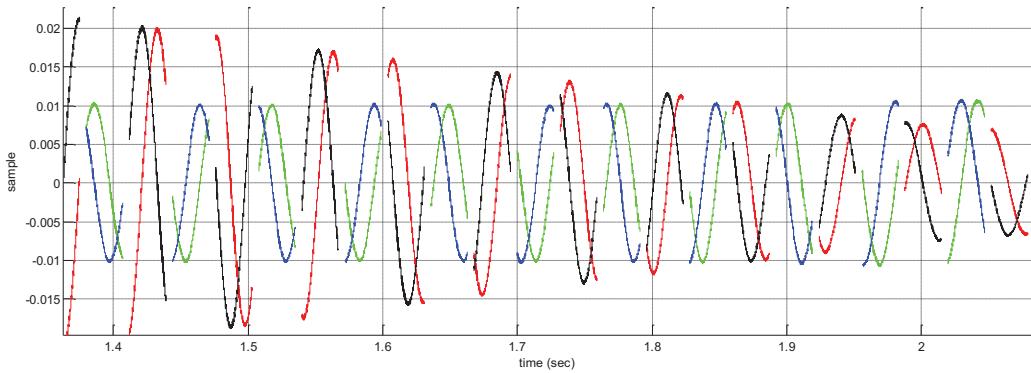
Figure 4-18 Plot of the real and imaginary components of each sample. Green and blue correspond to the reference frequency and red and black to the receiver components

Initially an approximate phase shift between the reference and the receiver channels can be determined by multiplying the signal in a reference channel by the conjugate of the interval of samples in the adjacent receiver channel. The angle of the complex valued signal provides an estimate of the difference phase that unfolds as the antenna is moved. Unwrapping is applied to the phase angle of these samples. The phase trajectory is plotted in Figure 5-19.
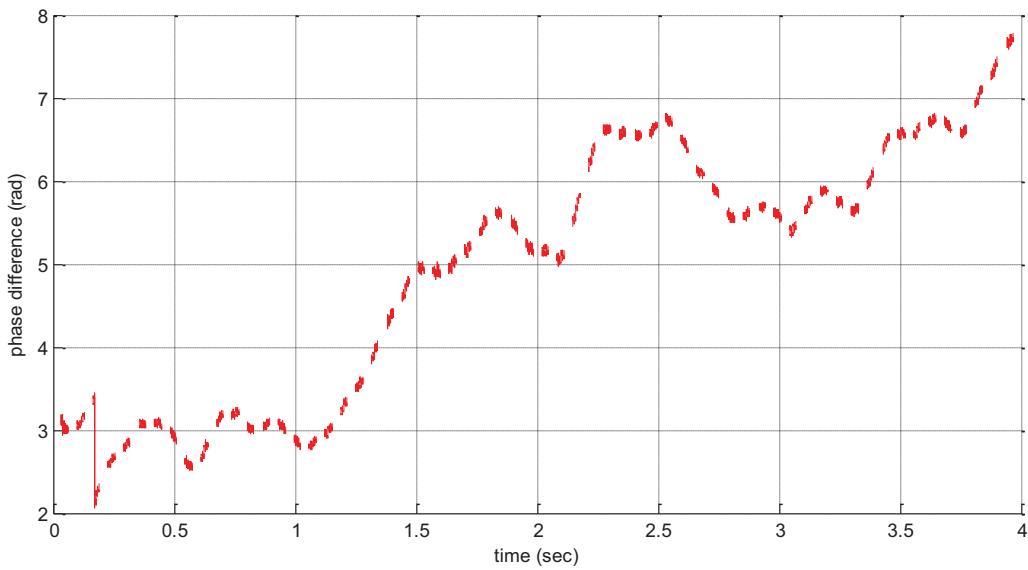
Figure 4-19 Approximate phase excursion as the antenna is moved.

Clearly the phase excursion in Figure 5-19 is not overly accurate but it does provide the experimenter a quick assessment of the quality of the data.   Other checks can be done such as stability of the signal s when the turntable is not moved. This is shown in Figure 4-20.
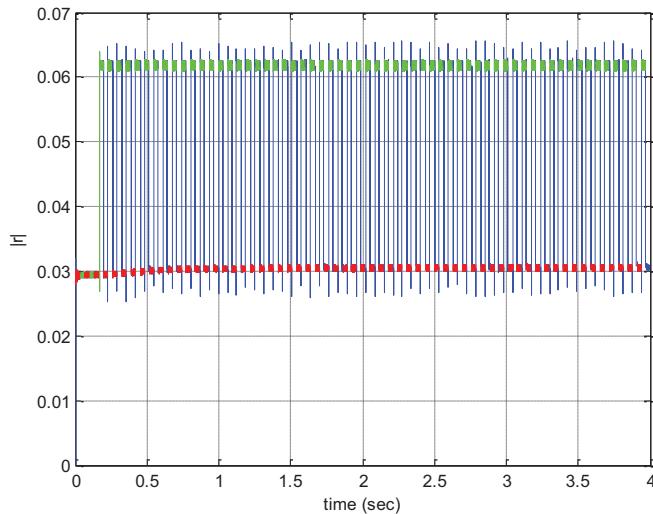
Figure 4-20 Response when the antenna is held constant position

Fig 4-21 shows the partitioning of the sampling with the red and green as the portions of the samples that for the reference and receiver channels. The blue samples are discarded as they correspond to the transition regions.
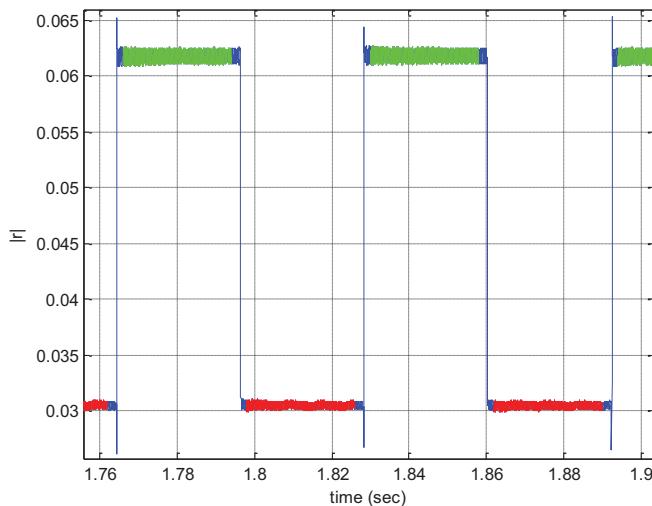
Fig 4-21 Detail of Figure 4-20

4.4 Simultaneous CV and LTE outputs and processing

Unfortunately, the USRP operation consumes significant bandwidth of the USB interface to the laptop. Likewise the webcam demands a near maximum transfer rate when the webcam is streaming video at the rate of more than 20 frames per second. The Windows operating system cannot accommodate two simultaneous USB streaming sources and therefore if the units are run at the same time then the operation will be jerky with unpredictable delays. While the CV can accommodate interruptions to handle the USRP operation, the USRP needs to be reset every time which causes excessive delays and the phase coherency will not be maintained.

Therefore it is necessary to use two laptops to accommodate both the webcam input and the USRP. The labtops are time synchronized with a common HW switch that is triggered at the start and end of the sampling. Once the raw samples have been stored they will be processed in parallel on the two laptops. An Ethernet connection between the two computers combines the processed CV and USRP data on a single computer. This is then fed into the overall SLAM algorithm.

At this stage, due to the interfacing issue, it is not possible to run the HSA hardware completely in real time. However, the processing of the raw samples is relatively fast such that the SLAM processing starts essentially as soon as the sampling is completed.
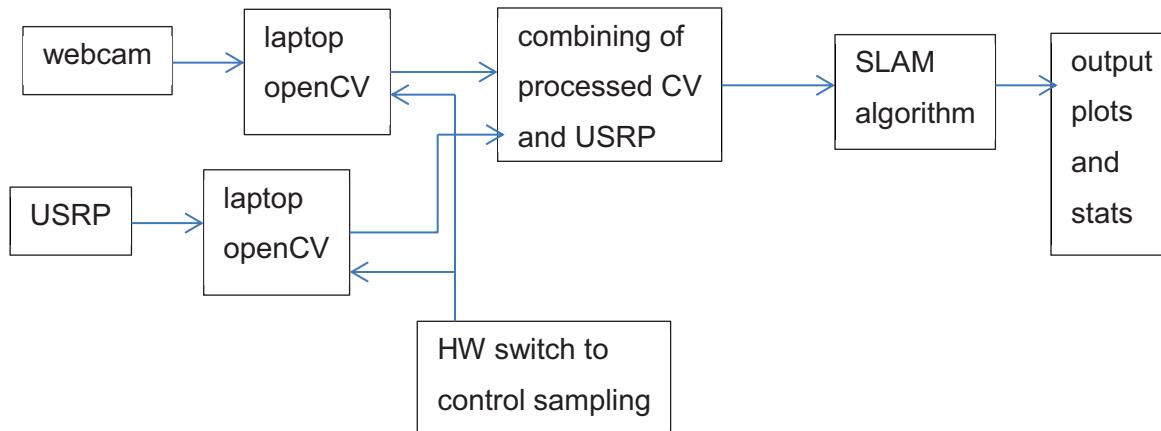
Fig 4-22 Overall flow of the data sampling and processing algorithm

4.5 Upgraded Experimental system based on NI RF PXI

An upgraded indoor localization experimental system based on wireless RF signal measurements is established. The hardware of the system mainly consists of NI RF PXI system, NI USRP, antennae, turntable, and switches. The software, based on LABVIEW, implements the configuration of the NI PXI system and USRP, control of the external devices, and synchronization of the overall system. RF signal measurements are acquired and streamed to the PXI system, where they are stored and can be used to apply the data post-processing. The following resources are needed for the system setup:

- Hardware

  – NI PXIe-1075 (18-Slot 3U PXI Express Chassis with AC - Up to 4 GB/s)

  – NI PXI-8135 (PXI Express Embedded Controller)

  – NI PXIe-5663E (6.6 GHz Vector Signal Analyzer (VSA) With RF List Mode)

  – NI PXIe-6555 (200 MHz Digital Waveform Generator, High Speed Digital IO (HSDIO)) with NI CB/SCB – 2162 single-ended digital I/O accessory

Generally, we call the combination of the above devices as PXI system, which is shown in figure 4-23.
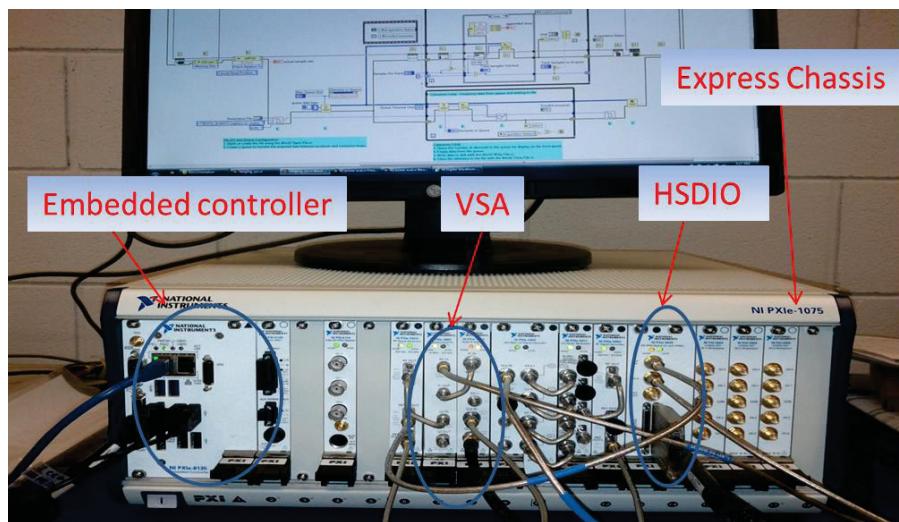
Figure 4-23: PXI system

  – NI USRP-2920 (50 MHz to 2.2 GHz Software Radio)

  – Turntable and controller

- Dipole LTE Antenna

- Coaxial SP4T switch

▪ Software

- LabVIEW 2012

- MathWorks MATLAB 2013a

The application of this experimental system is to do indoor wireless localization based on synthetic array (SA). Basically, in the experiment we need a receiver moving along a known trajectory while receiving wireless signals from access nodes (AN) and find its position relative to the AN.
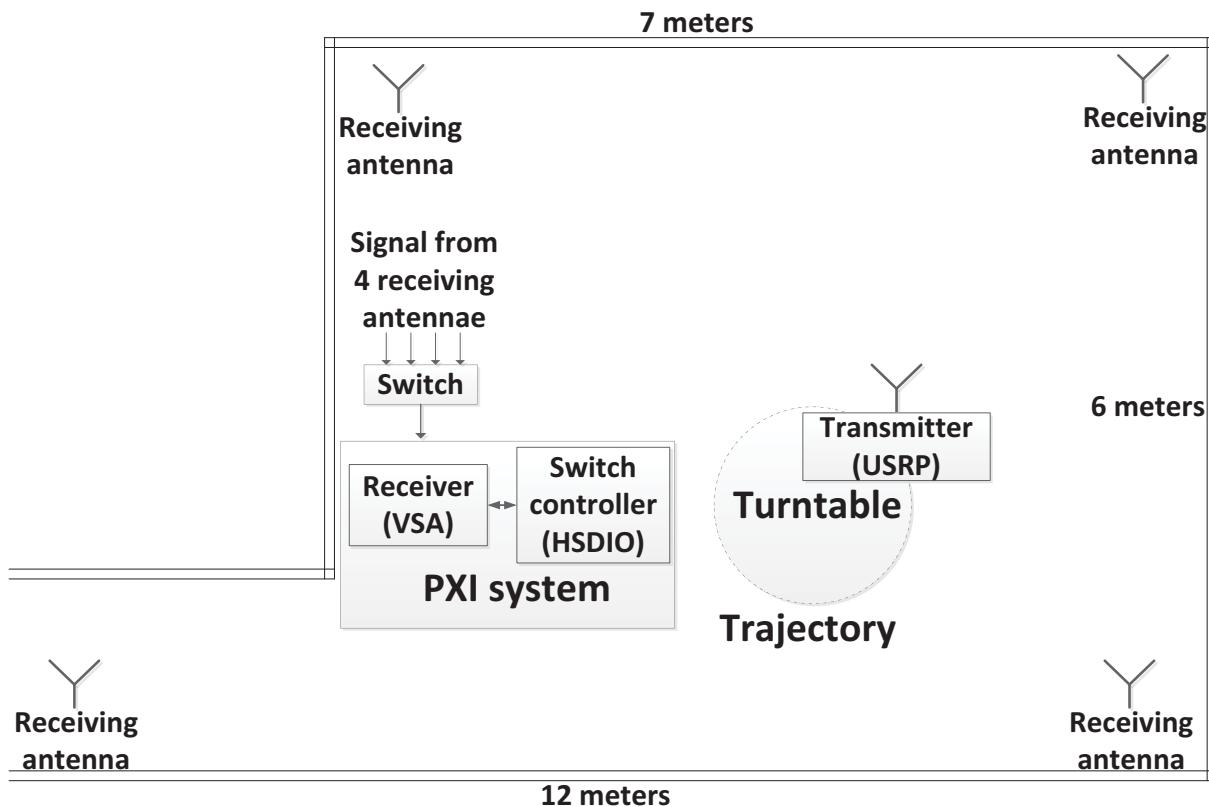
Figure 4-24: Overview of the indoor localization experimental system

Figure 24 shows the overview of the experimental system. The experimental room is at ENA401 at the University of Calgary. Basically, a transmitter is placed in a turntable moving in a circle. 4 receiving antennae are located at each corner of the room to receive signal from the transmitter. Then the 4 received signals will be scanned by a switch alternatively and fed to a receiver, where the signal is proceed to obtain the measurements for localization.

Figure 24 also shows the implementation of the system. Specifically, the receiver is implemented by USRP. Figure 3 shows the USRP and the transmitting antenna on
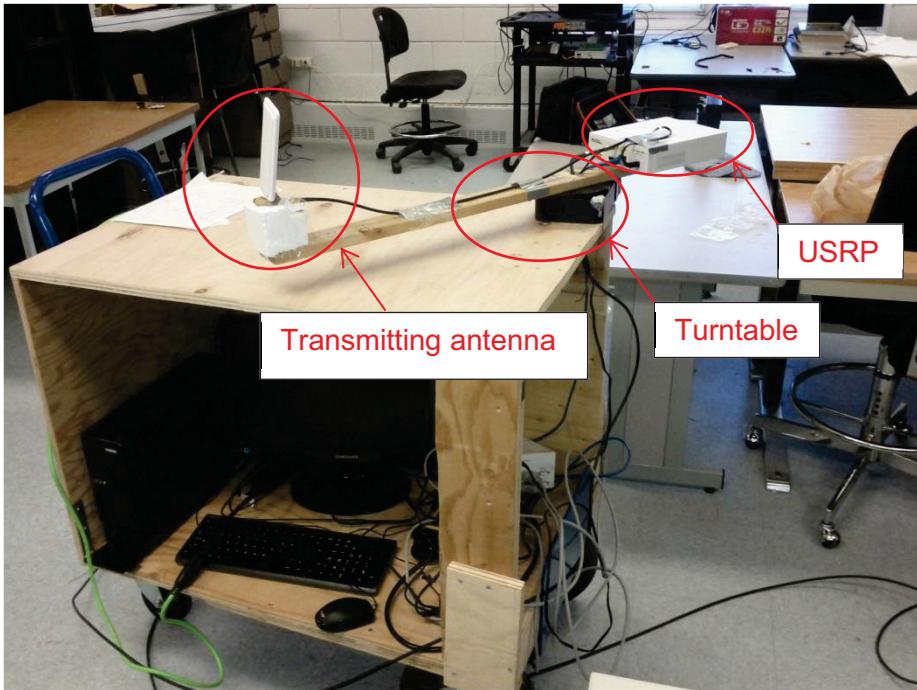
Figure 4-25: URSP and transmitting antenna on the turntable

the turntable. Shown in figure 26 is the receiving antenna in the corner of the room.

Figure 4-26: Receiving antenna placed in the corner of the room
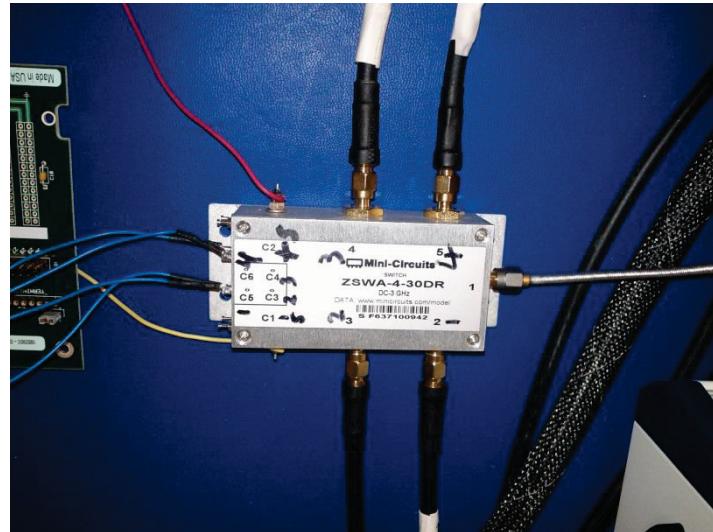
Figure 4-27: Switch used to scan signals from receiving antennae

The switch is controlled by HSDIO and the accessory board as shown in figure 4-28.
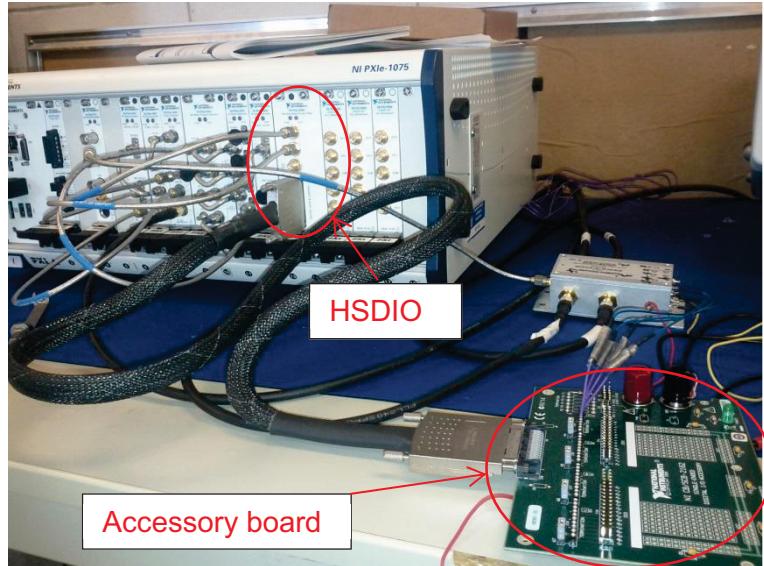
Figure 4-28: HSDIO and accessory board to control switch

The receiver is implemented using VSA in PXI system, as shown in figure 4-23.
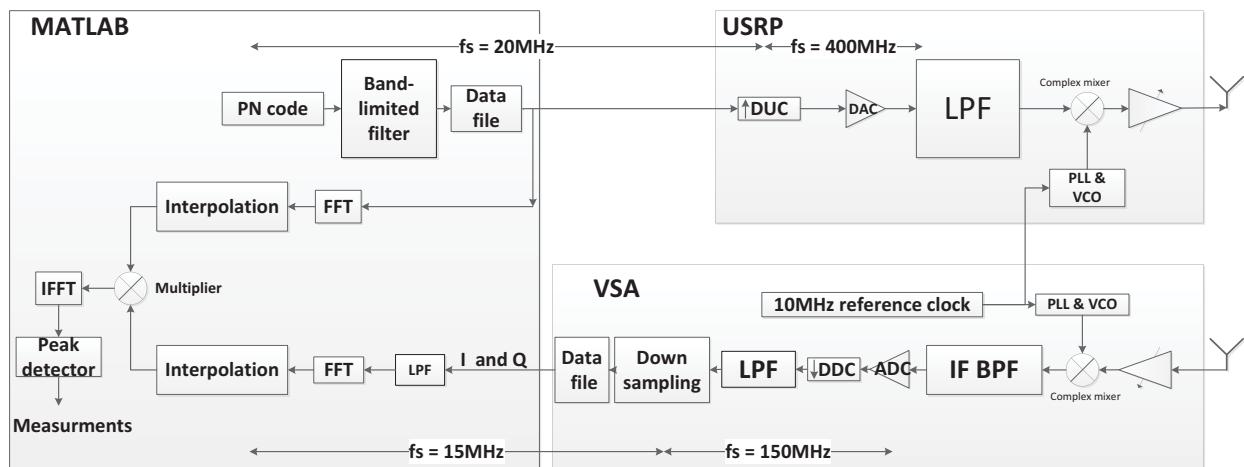
Figure 4-29: Signal processing functional blocks of the experimental system

Figure 4-29 describes the signal processing functional blocks of the experimental system. Generally, a bandlimited pseudorandom-noise (PN) code is generated and stored in a file using MATLAB. USRP reads the file to fetch baseband waveform, and up-converts, interpolates, low pass filters, and mixes the signal to RF for transmitting. In the receiver part, the VSA receives the RF signal, then down-converts, band pass filters and digitizes the signal. Decimation is then applied to the signal and the received baseband waveform is stored in a file for further processing. In the post-processing, the received signal is dispread using the local PN code, from which the amplitude, phase, time of arrival (TOA) measurements can be obtained for the localization algorithm.

Unlike the devices realised by NI which can be control by LABVIEW directly, the way to control the switch and turntable need to be given more thoughts, so as to make the system integration easier and more efficient. The HSDIO provides flexible software define way to generate digital waveform. Along with accessory board, HSDIO can control the switch through LABVIEW. Figure 8 shows the HSDIO accessory board and switch as well as the connection
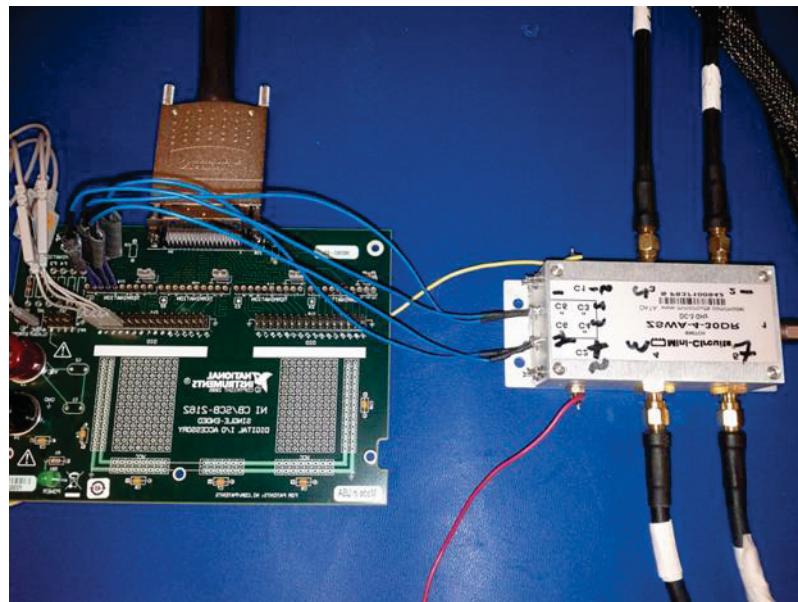
Figure 4-30: HSDIO accessory board and switch

between them. For the turntable, its controller can be programed using LABVIEW based driver,
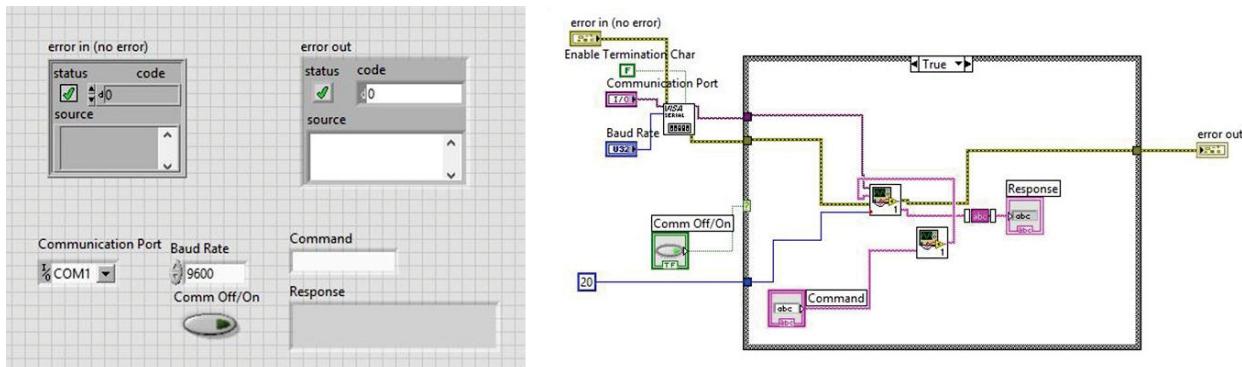
Figure 4-31: Turntable LABVIEW based drive

as shown in figure 4-31.

To sum up, all the hardware involved in the system can be control by LABVIEW, which makes the integration of the system using one single host program possible. Figure 4-32 shows the
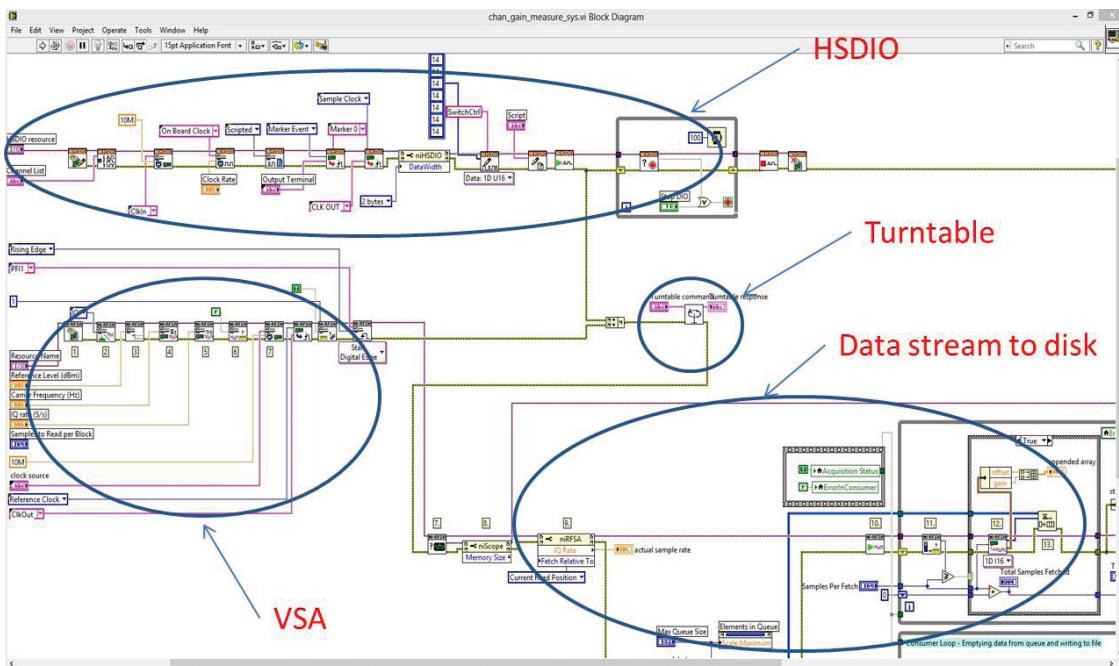
Figure 4-32: Overview of the block diagram of the host program

overview of the block diagram of the host program. Figure 11 shows the Front panel of the host
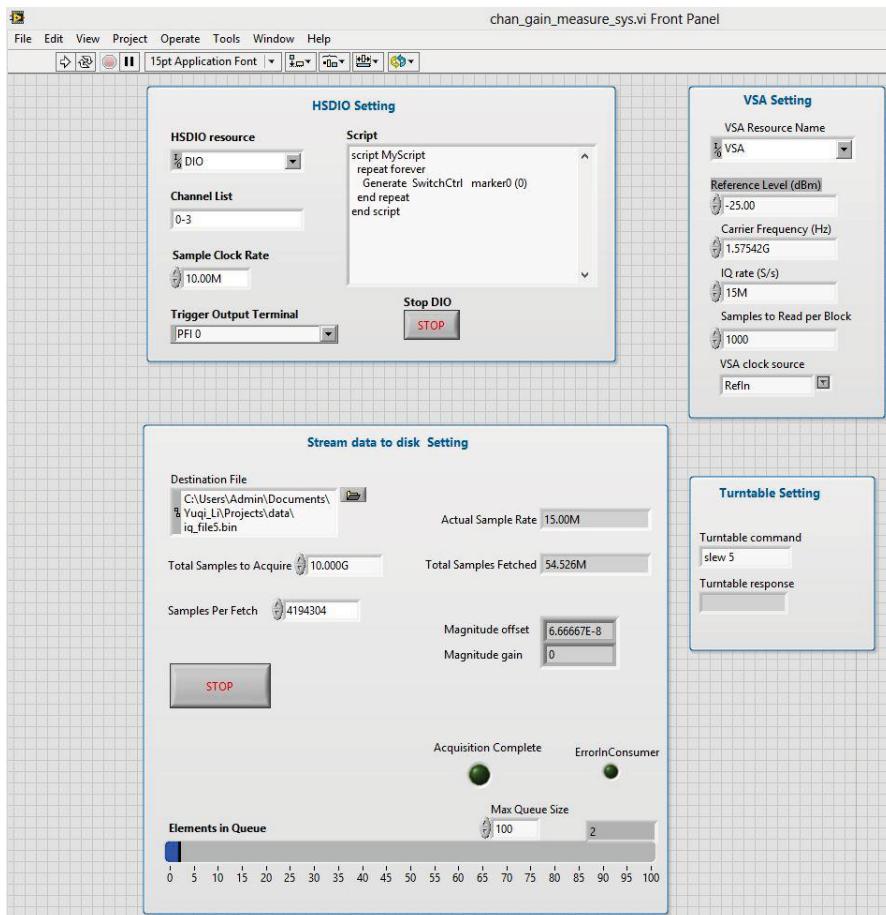
Figure 4-33: Front panel of the host program

program. In the experiment, we just need to configure the setting of each module in the program front panel, and simply press start button. Then the turntable, switch and receiver will start working in a synchronized way, and the received baseband waveform will be recorded in a file specified by the program for further processing.

FREQUENCY SYNCHRONIZATION OF TRANSMITTER, RECEIVER AND SWITCH CONTROLLER

Frequency synchronization plays an essential role in the integration of such a system involving multiple hardwires. Figure 4-34 explain the frequency synchronization scheme of the system.
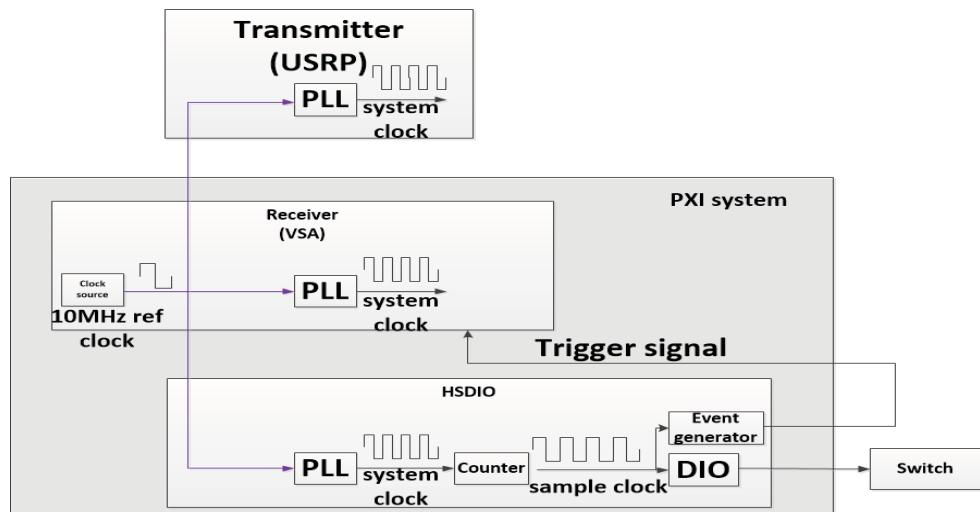
Figure 4-34: Frequency synchronization scheme of the system

As shown, USRP, VSA and HDSIO share a same reference clock of 10MHz in VSA, and generate the respective system clock using respective PLL. Then each system clocks will be used to generate other clocks to drive the modules in this device. The synchronization of transmitter and receiver in a hardwired connection way is an option to make the synchronization easier in the first place, and will be replaced by some other algorithm in the receiver later.

To implement above scheme, features of VSA, USRP and HSDIO are utilized in the design. Generally, VSA can export internal reference clock to the ports in its panel. As shown in figure
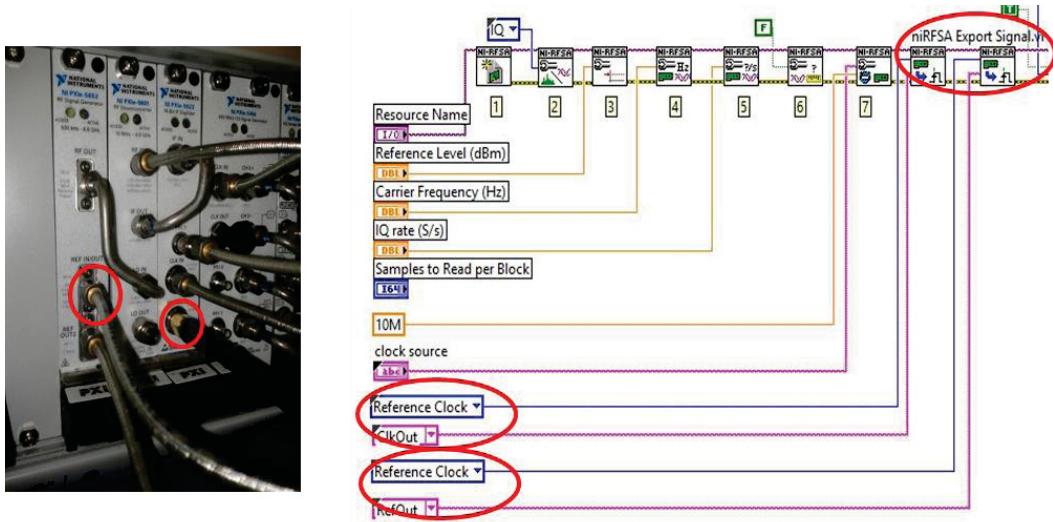
Figure 4-35: VSA reference clock outputs and configuration

13, through the configuration in the program, the internal reference clock of VSA are available in the 2 ports highlighted in the figure. USRP and HSDIO can accept external clock source as reference clock, as shown in figure 14 and figure 15. So we can simply connect their reference

*Figure 4-36: USRP reference clock input and configuration*
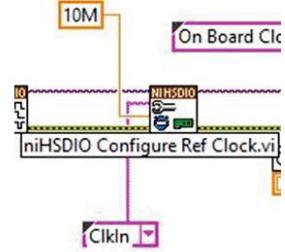
Figure 4-37: HSDIO reference clock input and configuration

clock input port to the clock source port in VSA panel and finish the setting for frequency synchronization.

EVENT SYNCHRONIZATION OF RECEIVER AND SWITCH CONTROLLER

Event synchronization of the receiver and switch controller must be performed to identify the signal from different channels. Figure 38 gives the event synchronization scheme.
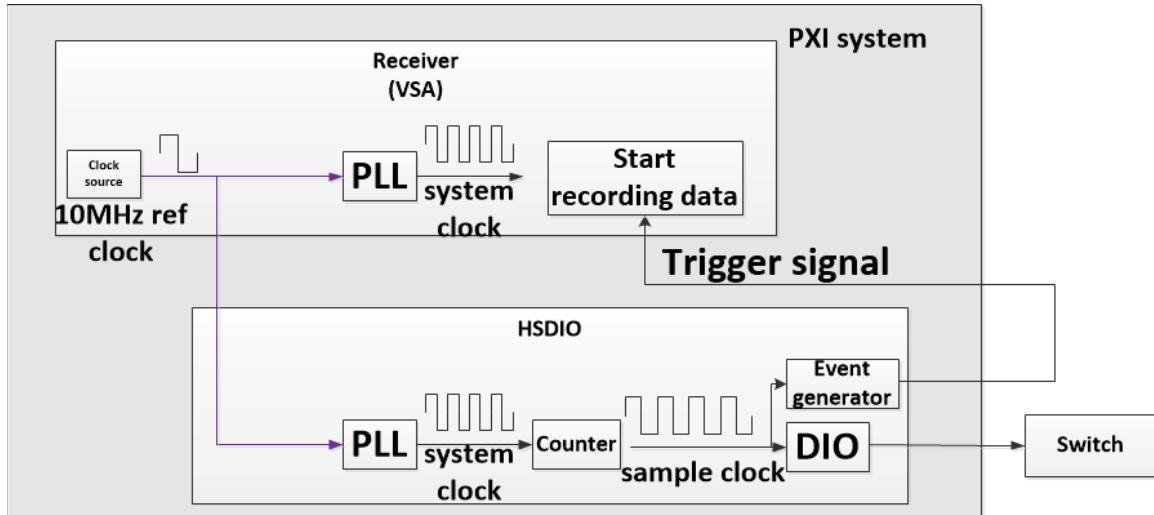
Figure 16: Event synchronization scheme of receiver and switch controller

Basically, HSDIO uses its sample clock to generate synchronized switch control signal and event trigger signal. While the switch control signal is fed to the switch, the event trigger signal is routed to the VSA to trigger VSA start recording data.

Features of HSDIO and VSA are used to implement the scheme above. HSDIO is able to generate digital I/O waveforms and events signal (called marker event) in specified patterns by script, as shown in figure 39.
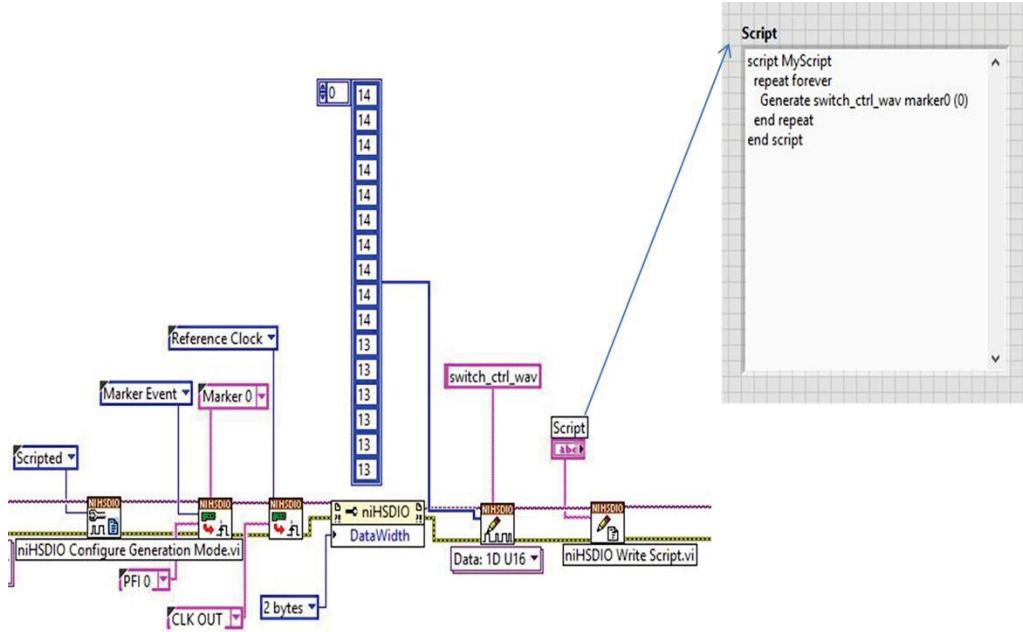
Figure 4-39: Configuration of HSDIO to generate digital I/O waveform and event trigger signal

The digital I/O waveforms then will be exported to I/O at the accessory board to control the switch, while event signal will be export to the port in HSDIO panel shown in figure 4-40.
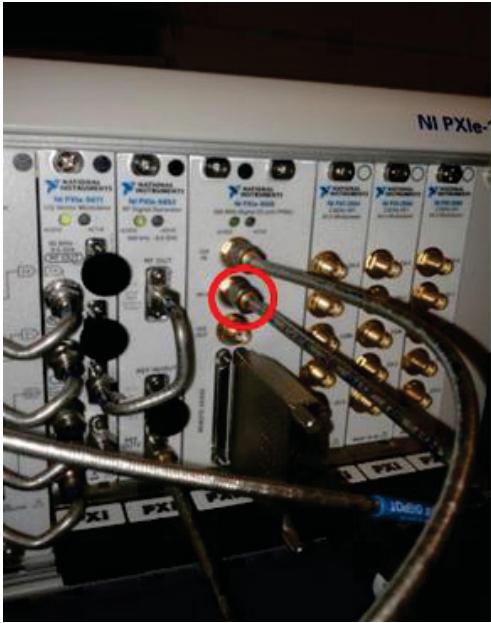
Figure 4-40: HSDIO event signal output and configuration

Since VSA is able to accept external event trigger input, as shown in figure 4-41, to trigger the data recording, we can simply connect this input port to the event trigger source in HDSIO port and finish the event synchronization setting.
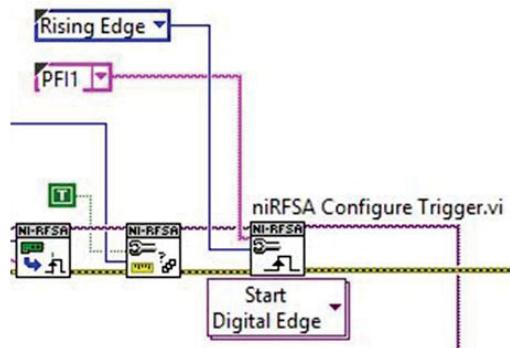

Figure 4-41: VSA event trigger input and configuration

Figure 20 shows the generated switch control signal and VSA event trigger signal capture by logic analyzer. As can be seen, the VSA event trigger signal can mark the beginning of the
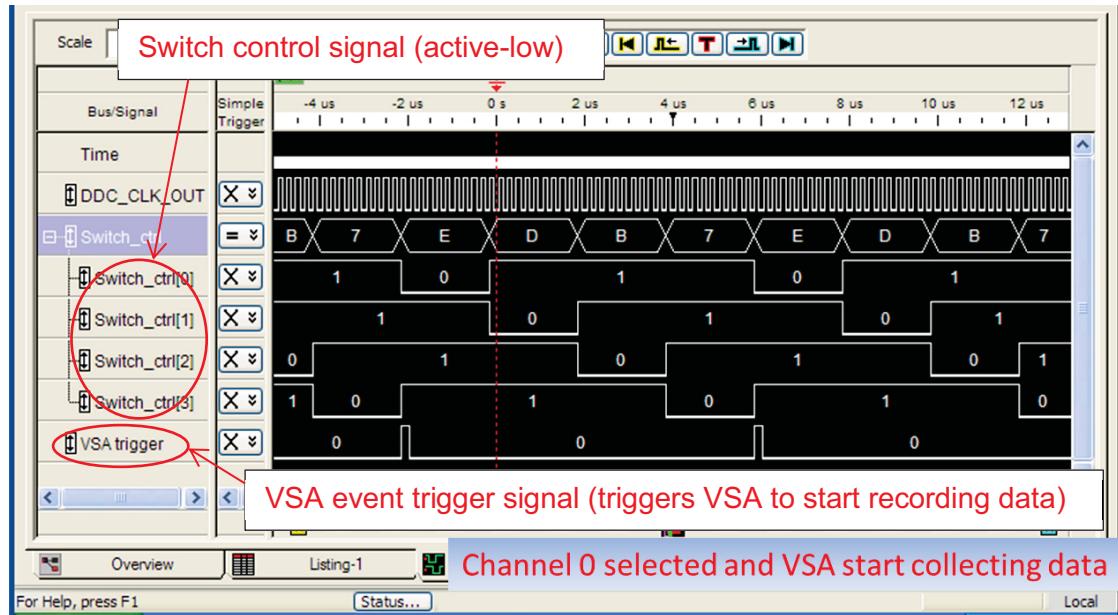
Figure 4-42: Switch control signal and VSA event trigger signal generated

channel 0 data recording. Figure 21 shows the baseband waveforms recorded in file. Obviously, we can separate the date of each channel very easily because the beginning of the date record corresponds to that of the first channel.
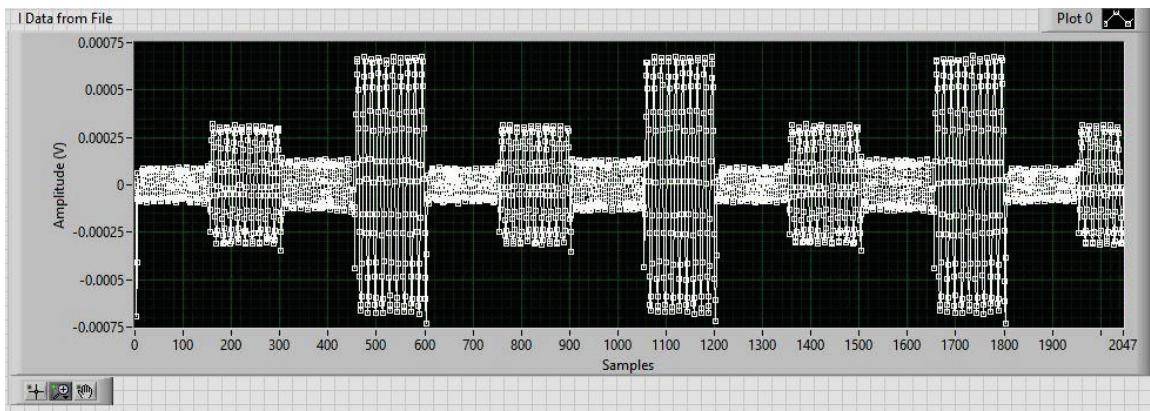

Figure 4-42: Waveform record of 4 channels

Experimental results

Figure 4-43 shows the measurements and location results of an experiment. In this experiment, only 1 receiving antenna is used and is placed in the position highlighted by the red star, while
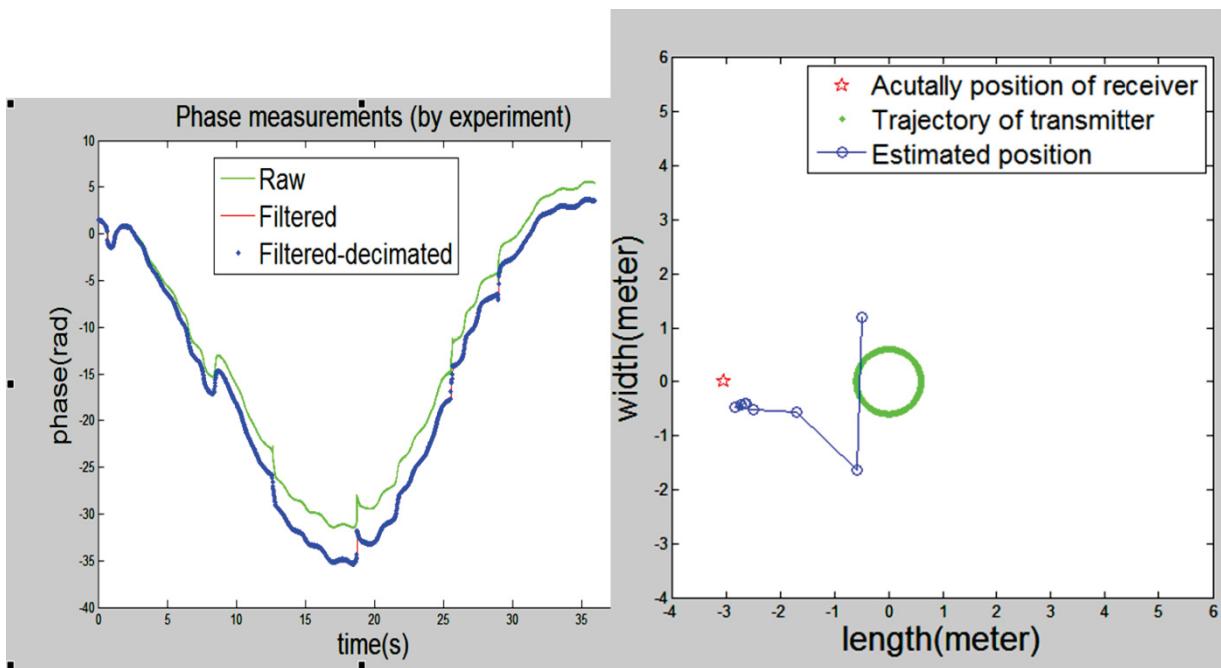
Figure 4-43: Waveform record of 4 channels

the transmitter is moved along a circle represented by green circle in right graph. The phase measurements, after removing circle clip, are shown in the left graph with green dots. We then apply localization algorithm to phase measurements, and the blue circles in the right graph show how the position estimate converges, which tells that the position estimated by receiver is very close to the actual position.

5. Discussion

The initial objective of this research was to develop elements of a system that could use the LTE wireless signals of opportunity for self geo-location (SGL) of a handheld navigation device (HND). This implied that the HND was not registered with the wireless system. The issues of not being able to assume that the LTE sources were in known locations or time synchronized implied that some form of SLAM processing was necessary. It was determined that provided that it could be assumed that the LTE access nodes were stationary and frequency synchronized (not code or offset synchronized) then SLAM processing would be able to simultaneously determine where the LTE access nodes are and their offsets in time delay as well as the trajectory of the HND relative to the LTE network of access points. SLAM also provided the most general framework for implementing the synthetic array component. Extensive details of the SLAM algorithm implementation and analysis is contained in part B of this final report.

While the multipath of the LTE emissions can be modelled such that the SLAM algorithm can to some degree average out the variances of the errors of the HND trajectory, this is shown to be the difficult component of achieving a sub-meter accuracy. Hence CV observables were added into the mix of measurement constraints. These were shown to be significant in terms of the overall SLAM in that CV positioning is highly accurate over short trajectory lengths but subject to drift. Contrary wireless with multipath is rather inaccurate due to multipath but not subject to the same drift as with the CV. Hence CV and wireless observables are complementary. Of course other observables can be used such as inertial but this was beyond the scope. Inertial will have the same issues as CV in terms of drift. However, as inertial drifts with time and spatial movements, CV only drifts as a result of nonlperfect calibration of the camera, image noise, non-ideal features and imager discretization. CV is not affected by time lapse. Again inertial can be construed as being complementary to both CV and wireless. It should therefore be included in the mix of observables.

An emerging development that we did not have bandwidth to cover is that of cooperative positioning (CP) which fits very nicely within the overall SLAM framework. This is the joint

location of several mobile nodes simultaneously which share information. CP requires some form of messaging between the mobile nodes or messaging to a central node. It is easily demonstrated in the SLAM formation given in part B that CP can significantly multiply the positioning information that is mutually available reducing the variance of the desired variables associated with the trajectory of the mobile units.

Future work would focus on the combining of the three sensor types, wireless, CV and inertial into a single instrumented experimental testbed. While experimental apparatus was developed for the wireless and CV, these need to be combined in a more unified way such that the relative significance of the measurement modalities can be assessed.

Main References Used

"LTE The UMTS Long Term Evolution 2nd edition", Stefanie Sesia, Issam Toufik, Matthew Baker.

3GPP TS 36.211: "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation".

3GPP TS 36.212: "Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding".

3GPP TS 36.213: "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures".

3GPP TS 36.214: "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer – Measurements".

S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics", MIT press, 2006

M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, " A tutorial on particle filters for online Nonlinear/Non-Gaussian Bayesian tracking", IEEE transactions on signal processing Vol.50, No.2 Feb 2002, pp.174-188

H. VanTrees, "Optimum array processing", John Wiley, 2002

S. Kay, "Fundamental of statistical signal processing – Estimation theory", vol. 1, Prentice Hall 1998

S. Kay, "Fundamental of statistical signal processing – Detection theory", vol. 2, Prentice Hall 1998

http://www.endruntechnologies.com/frequency-standard-oscillators.htm

Pablo Pedregal, "Introduction to optimization", Springer volume 46, 2003

Durrant-Whyte, H.; Bailey, T.; , "Simultaneous localization and mapping: part I," Robotics & Automation Magazine, IEEE , vol.13, no.2, pp.99-110, June 2006

R. Smith and P. Cheesman, "On the representation of spatial uncertainty," Int. J. Robot. Res., vol. 5, no. 4, pp. 56–68, 1987.

H.F. Durrant-Whyte, "Uncertain geometry in robotics," IEEE Trans. Robot. Automat., vol. 4, no. 1, pp. 23–31, 1988.

N. Ayache and O. Faugeras, "Building, registrating, and fusing noisy visual maps," Int. J. Robot. Res., vol. 7, no. 6, pp. 45–65, 1988.

J. Crowley, "World modeling and position estimation for a mobile robot using ultra-sonic ranging," in Proc. IEEE Int. Conf. Robot. Automat., 1989, pp. 674–681.

R. Chatila and J.P. Laumond, "Position referencing and consistent world modeling for mobile robots," in Proc. IEEE Int. Conf. Robot. Automat., 1985, pp. 138–143.

M. Csorba, "Simultaneous Localisation and Map Building," Ph.D. dissertation, Univ. Oxford, 1997.

S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots," Machine Learning, vol. 31, no. 1, pp. 29–53, 1998

F. van Diggelen, "Indoor GPS theory and implementation:' Proc IEEE Position Location and Navigation Symp., pp. 240-247, April 2002

M. lrsigler, B. Eisrfeller, "Comparison of multipath mitigation techniques with consideration of future signal structures," Proc. ION GPS/GNSS, pp. 2584-2592, Sept. 200

M. Fiala, "Designing Highly Reliable Fiducial Markers," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.32, no.7, pp.1317-1324, July 2010.

S.S. Chawathe, "Marker-Based Localizing for Indoor Navigation," Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, vol., no., pp.885-890, Sept. 30 2007-Oct. 3 2007.

T. Manabe, S. Yamashita, T. Hasegawa, "On the M-CubITS pedestrian navigation system," Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE , vol., no., pp.793-798, 17-20 Sept. 2006

H. Lim, Y. S. Lee, "Real-time single camera SLAM using fiducial markers," ICCAS-SICE, 2009, vol., no., pp.177-182, 18-21 Aug. 2009.

C. Golban, C. Mitran, S. Nedevschi, "A practical method for ego vehicle motion estimation from video," Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on , vol., no., pp.87-94, 27-29 Aug. 2009.

F. van Diggelen, "Indoor GPS theory and implementation:' Proc IEEE Position Locotion and Navigation Symp., pp. 240-247, April 2002

M. lrsigler, B. Eisrfeller, "Comparison of multipath mitigation techniques with consideration of future signal structures," Proc. ION GPS/GNSS, pp. 2584-2592, Sept. 2003

M. Fiala, "Designing Highly Reliable Fiducial Markers," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.32, no.7, pp.1317-1324, July 2010.

S.S. Chawathe, "Marker-Based Localizing for Indoor Navigation," Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, vol., no., pp.885-890, Sept. 30 2007-Oct. 3 2007.

T. Manabe, S. Yamashita, T. Hasegawa, "On the M-CubITS pedestrian navigation system," Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE , vol., no., pp.793-798, 17-20 Sept. 2006

H. Lim, Y. S. Lee, "Real-time single camera SLAM using fiducial markers," ICCAS-SICE, 2009, vol., no., pp.177-182, 18-21 Aug. 2009.

C. Golban, C. Mitran, S. Nedevschi, "A practical method for ego vehicle motion estimation from video," Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on , vol., no., pp.87-94, 27-29 Aug. 2009.

Y. Ma, S. Soatto, J. Kosecka, S.Sastry, An Invitation to 3D vision, from images to geometric models, Springer 2006

K. Nicholson Linear Algebra with applications, 5th ed McGraw Hill 2006